

# Spécifications techniques : Qwenta

Le but de ce document est de définir les spécifications techniques générales de l'outil  
"Menu Maker by Qwenta"

Version 1.0	Auteur Webgencia	Date 23/09/2024	Approbation Qwenta
----------------	---------------------	--------------------	-----------------------

## I. Choix technologiques

État des lieux des besoins fonctionnels et de leurs solutions techniques :

<b><u>Besoin</u></b>	<b><u>Contraintes</u></b>	<b><u>Solution</u></b>	<b><u>Description Solution</u></b>	<b><u>Justifications</u></b>
<b>Page d'accueil (Landing Page)</b>	Doit comprendre efficacement l'outil et comment débiter avec la plateforme (création de menu)	<ul style="list-style-type: none"><li>• HTML</li><li>• CSS</li><li>• React</li></ul>	Création/intégration de la page web	<ul style="list-style-type: none"><li>- Le HTML et CSS pour la partie statique</li><li>- React pour la partie dynamique et modulable.</li></ul>
<b>Connexion User</b>	<ul style="list-style-type: none"><li>• Doit pouvoir se connecter à son espace</li><li>• Espace sécurisé par un login/mdp</li></ul>	<ul style="list-style-type: none"><li>• Bcrypt (pour le hachage du mdp avant stockage bdd)</li><li>• JWT pour la génération de token d'authentification</li><li>• HTML</li><li>• CSS</li><li>• React</li><li>• API</li><li>• MongoDB</li></ul>	L'api va permettre de communiquer avec la BDD user afin d'agir sur celle-ci, chiffrer les données et authentifié/autorisé l'utilisateur. React/html/css pour l'affichage du front qui enverra/recevra les données avec le backend.	Outils fiables et flexibles dans leurs utilisations, intégrations (environnement React). Chiffrement des informations sensibles

<b>Gestion des modales (création menu)</b>	Le user doit pouvoir réaliser toutes les étapes de création de son menu sans perdre en qualité d'expérience user	React-modal React HTML CSS	Il s'agit d'une bibliothèque populaire utilisée pour créer des modales dans les applications React	Performant et très flexible (import dans les différents composants, etc..) React-modal est une biliot
<b>Création de menus</b>	Le user créer son menu : <ul style="list-style-type: none"> <li>• Choix des catégories</li> <li>• Choix des plats</li> <li>• Ajout d'informations (images, textes, titre et prix)</li> </ul>	<ul style="list-style-type: none"> <li>• React</li> <li>• React-FileReader</li> <li>• API</li> <li>• HTML</li> <li>• CSS</li> </ul>	L'api pour les opérations CRUD Le file Reader pour l'ajout/preview de l'image React pour la partie dynamique	<ul style="list-style-type: none"> <li>• Très flexible avec sa gestion des composants</li> <li>• Performance</li> </ul>
<b>Personnalisation des menus (Style)</b>	Le user sera en mesure de choisir ses couleurs et polices d'écritures selon une liste prédéterminée d'options	React style-components	Grâce à style-component, vous pouvez facilement modifier les styles de vos composants en fonction des choix de l'utilisateur.	Flexible et dynamique
<b>Exportation PDF du menu</b>	L'utilisateur sera en capacité d'extraire son menu sous le format pdf. Doit aussi pouvoir prévisualiser son menu.	React-pdf React-FileReader	Permet la création du pdf et l'extraction de celui-ci	Conçu exclusivement pour React donc flexible, dynamique et performant.
<b>Demande d'impression du menu</b>	L'utilisateur pourra commander l'impression de son menu via Qwenta ?	<ul style="list-style-type: none"> <li>• CSS pour la fonction d'impression Print</li> <li>• React Router pour le composant Link</li> </ul>	Un peu comme l'exportation PDF, on exportera les éléments sélectionnés dans le css pour l'impression et sera envoyer au back-office	On a choisi le composant link à une simple balise <a> au cas où nous devrions réutiliser ce composant ailleurs sur le site Web.
<b>Dashboard</b>	L'utilisateur doit pouvoir avoir accès à un tableau de bord regroupant les	<ul style="list-style-type: none"> <li>• API</li> <li>• React</li> </ul>	API pour l'affichage et Mise à jour en temps réel des informations.	API + Framework (React/Express) sont largement suffisant et performant pour

	informations sur la création de menus, leur diffusion/impression et 3 derniers articles Qwenta		React pour l'affichage des redirections vers les 3 services concernés	ce besoin
<b>Visualisation Menus précédents</b>	L'utilisateur doit pouvoir visualiser et accéder à toutes ses créations antérieures	<ul style="list-style-type: none"> <li>• API</li> <li>• React</li> <li>• React-modal</li> </ul>	<ul style="list-style-type: none"> <li>• Affichage de tous les menus déjà créés par l'utilisateur</li> <li>• Affichage de la date de création des menus</li> <li>• Possibilité de modifier/Supprimer</li> <li>• Possibilité de créer un menu directement via cet interface de présentation (mise à jour en temps réel sans refresh de la page)</li> </ul>	Les opérations CRUD seront nécessaires à la bonne réalisation des opérations demandées. Les composants de la création de menus seront utilisés ici.
<b>Informations utilisateurs</b>	L'utilisateur doit pouvoir modifier ses informations user	<ul style="list-style-type: none"> <li>• API</li> <li>• React</li> <li>• React-modal</li> </ul>	<ul style="list-style-type: none"> <li>• Affichage des informations utilisateurs</li> <li>• Modification des données users</li> <li>• Ajout/Suppression de plusieurs adresses email différentes</li> </ul>	Les opérations CRUD seront nécessaires à la bonne réalisation des opérations demandées.
<b>Tarification</b>	L'utilisateur accède aux tarifs du menu maker dans un nouvel onglet	<ul style="list-style-type: none"> <li>• HTML</li> <li>• CSS</li> <li>• React</li> </ul>	<ul style="list-style-type: none"> <li>• Affichage des informations de tarification</li> <li>• S'ouvrira dans un nouvel onglet</li> </ul>	Le HTML et CSS pour la partie statique et le JS pour la partie dynamique.
<b>Branding Restauration</b>	L'utilisateur devra ajouter/supprimer	<ul style="list-style-type: none"> <li>• API</li> <li>• React-FileReader</li> <li>• React-StyleCo</li> </ul>	<ul style="list-style-type: none"> <li>• Affichage de l'identité visuelle du restaurant</li> </ul>	Les opérations CRUD seront nécessaires à la bonne réalisation

	r/modifier son logo et ses couleurs de bases	mponent	(logo / couleurs) <ul style="list-style-type: none"> <li>• Ajout/Suppression/Modification du logo</li> <li>• Modification des couleurs de base</li> </ul>	des opérations demandées. Le style-component permettra une personnalisation css du restaurant Le FileReader pour la gestion du logo (image)
<b>Information légales</b>	L'utilisateur doit avoir accès aux informations légales de Menu-Maker	<ul style="list-style-type: none"> <li>• HTML</li> <li>• CSS</li> <li>• JS React-modal</li> </ul>	<ul style="list-style-type: none"> <li>• Affichage des informations légales</li> <li>• S'ouvrira dans une modale</li> <li>• Mention "Tous droits réservés" figurera sur toute les pages</li> </ul>	Le HTML et CSS pour la partie statique et le JS pour la partie dynamique. React-modal pour la gestion de la modale
<b>API deliveroo</b>	L'utilisateur doit pouvoir exporter son menu en un clic vers l'application Deliveroo.	<ul style="list-style-type: none"> <li>• API Deliveroo</li> <li>• React</li> </ul>	<ul style="list-style-type: none"> <li>• Connexion à l'API Deliveroo</li> <li>• Exportation du menu (requête POST)</li> <li>• Redirection vers l'appli Deliveroo lors du clic sur l'encart</li> </ul>	L'exportation du menu d'un restaurateur vers l'application Deliveroo en un clic nécessite une intégration entre notre application et l'API de Deliveroo.
<b>API Instagram</b>	L'utilisateur doit pouvoir exporter son menu facilement vers Instagram	<ul style="list-style-type: none"> <li>• API Instagram</li> <li>• React</li> </ul>	<ul style="list-style-type: none"> <li>• Connexion à l'API Instagram</li> <li>• Générer image carrés</li> <li>• Exportation et Redirection vers l'appli Instagram la suite du processus</li> </ul>	Cette solution permet au restaurateur de générer facilement des images carrées de son menu, prêtes à être partagées sur Instagram. Même si l'intégration directe avec Instagram est limitée pour les comptes personnels, cette approche

				manuelle assure une expérience fluide
--	--	--	--	---

## II. Choix Front End

### 1. Quel langage pour le Front ?

Le langage utilisé pour le Front-End sera du JavaScript avec le framework React.

Il s'agit du langage le plus répandu dans le développement Web et React, avec sa modularité/réutilisabilité, est un outil puissant pour construire des composants/interfaces utilisateurs.

## III. Choix Back End

### 1. Quel langage pour le serveur ?

Nous utiliserons NodeJs / Express.

Justif a mettre facon gepetto

### 2. A-t-on besoin d'une API ?

Ce projet va utiliser 3 APIs :

- API REST pour communiquer avec les bases de données (auth/ gestion des menus/ users..)
- API Instagram
- API Deliveroo

### 3. Base de données choisie :

Nous utiliserons la Base de données NoSQL Mongodb.

Il a principalement été choisi pour sa flexibilité (modèle de données orienté documents) et sa performance (lecture et écriture rapides, surtout lorsque les données sont volumineuses et que

les structures de requêtes ne sont pas trop complexes). Enfin sa scalabilité nous a convaincu de le sélectionner.

#### IV. Préconisations concernant le domaine et l'hébergement

##### **1. Nom du domaine.**

Partant du principe que Qwenta n'a pas déjà réservé un nom de domaine, il est possible de réserver le ou les noms de domaines suivants :

- Menu-maker.fr (disponible à l'achat)
- Menu-maker.com (disponible à l'achat par un tiers)

NB: Menu-maker.net (site chinois) est déjà réservé mais, à la vue de son contenu, nous ne pensons pas qu'il puisse représenter une menace en termes de cannibalisme.

##### **2. Nom de l'hébergement.**

Nous avons choisi de sélectionner OVH Cloud pour la qualité de son service et surtout pour sa politique de confidentialité vis-à-vis des données hébergées.

##### **3. Adresses e-mail.**

En fonction du nom de domaine choisi par Qwenta, nous pouvons prévoir plusieurs adresses emails afin d'optimiser la communication entre les différents services de Menu Maker :

- Contact@... (Boite mail générique)
- Rgpd@...

#### V. Accessibilité

##### **1. Compatibilité navigateur.**

Menu Maker sera compatible avec les dernières versions de ces navigateurs :

- Google Chrome
- Mozilla Firefox
- Safari
- Microsoft Bing

## **2. Types d'appareils.**

Menu Maker sera accessible et optimisé pour la version Desktop. Il n'a pas été prévu de versions mobiles et tablettes mais le projet sera conçu pour permettre le responsive dans le futur.

## **VI. Recommandations en termes de sécurité**

- Mises à jour régulière des bibliothèques, dépendances, frameworks ou tout autre plug-in en relation avec Menu-maker
- Encadrer rigoureusement les entrées de données users en proposant des validations de process sécurisées. Cela préviendra le spam ou tout autre injection non souhaitée.
- Respect des règles élémentaires de sécurité concernant le site Web et sa base de données (pas de données sensibles dans le code, chiffrement des données users/mdp, certificat SSL pour le https, 3d Secure pour le paiement)
- Réaliser un audit de sécurité une fois par an au minimum par une entreprise externe.
- Créer une classification d'erreurs afin d'identifier les erreurs de sécurité des autres erreurs afin d'optimiser le temps de traitement des problèmes liés à la sécurité (priorité extrêmement élevé).

## **VII. Maintenance du site et futures mises à jour**

### **1. Maintenance du site**

- Mise à jour régulière du site ainsi que ces composants (hebdomadaire ou quotidienne)
- Update du visuel/design du site (ajout/suppression contenu)
- Mise à jour de la documentation du site (Spécification techniques, fonctionnelles etc..)
- Vérification régulière du bon fonctionnement du site (tests des différentes fonctionnalités proposés par le site, stress tests et autres scénarios)

## 2. Mise à jour futures & évolution

- Prévoir l'implémentation du paiement en ligne (de type Stripe ou solutions propres aux banques)
- Mise en place de la version Responsive du Menu-maker
- Création/Vente de templates prédéfinis
- Forum ou espace communautaire afin que les users puissent échanger entre eux (sous modération)

## VIII. Composition équipe

- Lead Developpeur - Scrum Master (Sofiane)
- Product Owner (Sofiane)
- Testeur (Manon)
- Developpeur Frontend (Jonathan)
- Developpeur Backend (Christophe)
- Web Design (Valerie)
- Client Qwenta

## IX. Planning +