

Spécifications techniques

[Menu Maker by Qwenta]

Version	Auteur	Date	Approbation
1.0	Jonathan Huber	13/01/2026	John, Qwenta

I. Choix technologiques	2
II. Liens avec le back-end	3
III. Préconisations concernant le domaine et l'hébergement	3
IV. Accessibilité	3
V. Recommandations en termes de sécurité	3
VI. Maintenance du site et futures mises à jour	4

I. Choix technologiques

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

Besoin	Contraintes	Solution	Description de la solution	Justification (2 arguments)
Interface utilisateur moderne et dynamique	Application web, maintenable et évolutive	React	Librairie JavaScript pour créer une interface modulaire et réactive basée sur des composants	1) Standard du marché 2) Facilite la maintenance et l'évolution
Styliser l'interface utilisateur	Cohérence visuelle, maintenabilité, desktop only	Sass	Utilisation de CSS classique avec Sass pour bénéficier de variables, imbrication et mixins.	1) Sass améliore la maintenabilité et la cohérence des styles. 2) Contrôle total du design, facile à suivre avec la maquette.
Navigation entre les pages	Application monopage	React Router	Gestion du routage côté client sans recharge de page	1) Améliore l'expérience utilisateur 2) Solution officielle et mature
Affichage de modales (connection, création de menu)	Accessibilité.	React Modal	Composant léger pour afficher des fenêtres modales accessibles	1) Respect des bonnes pratiques d'accessibilité 2) Intégration simple avec React, léger.

Authentification par mail et gestion de l'adresse e-mail principale	Sécurité, gestion des comptes, persistance des données critiques	Firebase Authentication (Email Link)	Authentification par email Magic Link et modification de l'adresse e-mail principale gérées via Firebase Auth pour garantir sécurité et persistance	1) Scalable : Supporte plusieurs méthodes d'authentification 2) Google s'occupe de la sécurité, des tokens et de la protection contre les attaques classiques
Stockage des données applicatives	Temps réel, sécurité, modification, suppression et consultation des menus et informations secondaires	Cloud Firestore	Base NoSQL temps réel pour stocker menus et informations secondaires des utilisateurs, avec possibilité de créer, modifier et supprimer, accessible via SDK	1) Scalabilité automatique (augmentation du trafic et du volume de données) 2) Les règles de sécurité Firestore permettent de gérer précisément qui peut lire ou écrire quelles données.
Stockage des médias (logos, images)	Fichiers volumineux, accès sécurisé	Firebase Storage	Stockage cloud pour images et fichiers utilisateurs	1) Intégration native avec Firebase 2) Gestion fine des droits
Génération de menus PDF	Téléchargement et impression	@react-pdf/renderer	Génération d'un fichier PDF depuis un composant React dédié, basé sur les données du menu.	1) PDF standard et téléchargeable, conservant mise en page, typographie et qualité des images. 2) Simple à intégrer côté frontend.

Transformer le menu en image	- Rendu fidèle du menu (styles, couleurs, typographie) - Image exploitable pour publication Insta	html-to-image	Le composant Menu est capturé via html-to-image pour générer une image fidèle au rendu HTML/CSS, avec possibilité d'adapter le ratio selon le format souhaité.	1) Permet de générer une image au format Instagram, avec contrôle du ratio et de la résolution 2) Simple à mettre en place côté frontend
Partage du menu sur Instagram	Publier sur Instagram de façon automatique	- API Graph Instagram	Publication automatique des images du menu sur le compte Instagram professionnel du restaurant	1) Seule API officielle permettant de publier sur Instagram automatiquement 2) Évite la publication manuelle
Publication du menu sur Deliveroo	Automatisation	API Deliveroo	Les données du menu (catégories, plats, prix, descriptions, images) sont envoyées via l'API Deliveroo pour mettre à jour le menu sur la plateforme.	1) Automatise la mise à jour des menus sur Deliveroo. 2) Intégration directe et sécurisée avec la plateforme Deliveroo

Sécurisation des appels aux API externes	Clés sensibles, tokens d'accès, appels serveur requis	Firebase Cloud Functions	Les appels aux API Deliveroo et Instagram sont centralisés dans des Cloud Functions Firebase afin de sécuriser les clés, gérer les tokens et assurer un backend contrôlé.	1) Empêche l'exposition des clés côté client 2) Centralise la logique serveur pour toutes les intégrations
--	---	--------------------------	---	---

II. Liens avec le back-end

Le front-end React communique avec un backend serverless via des Cloud Functions Firebase.

Ce backend sert d'intermédiaire sécurisé entre l'application et les API externes (Deliveroo, Instagram), en gérant notamment la sécurisation des clés et des tokens.

- Quel langage pour le serveur ?

Node.js + TypeScript

- Les Cloud Functions sont écrites en Node.js/TypeScript.
- Il n'y a pas besoin de créer un serveur maison, Firebase s'occupe de l'exécution.

- A-t-on besoin d'une API ? Si oui laquelle ?

Oui.

- Api privée Deliveroo : publication automatique des menus.
- Api API Graph Instagram : publication automatique des images (comptes professionnels uniquement).
- API interne via Cloud Functions Firebase : exposée au front pour déclencher les actions serveur de manière sécurisée.

- Quelle base de données ?

Cloud Firestore (*NoSQL*)

- Stocke les menus et leurs métadonnées.
- Stocke les informations utilisateurs.

III. Préconisations concernant le domaine et l'hébergement

- Nom du domaine.
 - menumaker.qwenta.com (sous-domaine provisoire, en attente de validation)
- Nom de l'hébergement.
 - hébergé par Qwenta ou solution cloud externe type Firebase Hosting si Qwenta ne peut pas fournir l'hébergement
- Adresse e-mail.
 - contact@menumaker.qwenta.com

IV. Accessibilité et compatibilité

- Compatibilité navigateur.
 - Supporter les navigateurs modernes : Chrome, Firefox, Safari.
- Types d'appareils.
 - Desktop uniquement
- Accessibilité.
 - Navigation clavier complète (Tab, Escape, Enter)

- Contraste suffisant.
- Labels ARIA sur tous les éléments interactifs
- Focus visible sur tous les éléments cliquables
- Messages d'erreur accessibles
- Gestion du focus dans les modales (focus trap)

V. Recommandations en termes de sécurité

- Accès aux comptes : Firebase Authentication – Email Link + onAuthStateChanged pour sécuriser les pages sensibles.
- Données et fichiers : Firestore Security Rules pour restreindre l'accès aux données par UID ; Firebase Storage pour les images avec accès limité.
- Frontend et plugins : maintenir à jour React et dépendances, valider les uploads (type et taille), utiliser HTTPS.
- Monitoring et sauvegardes : activer les backups Firestore, surveiller logs et erreurs.

VI. Maintenance du site et futures mises à jour

- Maintenance corrective
 - Correction de bugs ou dysfonctionnements liés au frontend React.

- Surveillance de la compatibilité avec les nouvelles versions de React et des librairies utilisées (ex : react-modal).
- Maintenance évolutive
 - Ajout ou modification de fonctionnalités côté frontend : catégories, plats, menus, images.
 - Ajustement de l'interface utilisateur en fonction des retours restaurateurs.
- Mises à jour de sécurité
 - Mise à jour des dépendances React et des packages npm pour éviter les vulnérabilités.
 - Firebase s'occupe automatiquement de la sécurité côté backend et stockage, pas de serveur local à gérer.
- Support utilisateur
 - Assistance pour l'utilisation de l'interface (création de plats, menus, upload d'images).
 - Résolution des problèmes d'accès liés à l'authentification Firebase.
- Durée et fréquence
 - Contrat de maintenance annuel ou par forfait, avec un nombre d'heures dédié pour les corrections et évolutions.