

# Spécifications techniques

[Menu Maker by Qwenta]

Version	Auteur	Date	Approbation
1.0	Redwan Karoni	20/05/2024	Soufiane

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

## Choix technologiques

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

Besoin	Contraintes	Solution	Description de la solution
Authentification	Identification via E-mail	Firestore Authentication est un service d'authentification proposé par Firestore, qui assure une identification sécurisée des utilisateurs par le biais de leur adresse e-mail.	1)Firestore Authentication propose une intégration simple avec les applications web et mobiles, en faisant une solution idéale pour notre application Menu Maker. 2) C'est une technologie couramment utilisée et fiable, offrant une sécurité renforcée pour l'authentification des utilisateurs.

Création et sélection de catégories	L'utilisateur doit pouvoir via une modale être en mesure de choisir et de créer des catégories de plats	Utilisation de React Modal pour élaborer une fenêtre modale dédiée permettant à l'utilisateur de saisir le nom de la catégorie de plat. La modale doit comporter un champ pour l'entrée du nom et des boutons pour valider ou annuler la création de la catégorie	<p>1) React Modal est une bibliothèque React facile à utiliser et flexible, offrant une expérience utilisateur cohérente en affichant le contenu modale de manière soignée.</p> <p>2) Grâce à React Modal, nous pouvons facilement contrôler l'ouverture, la fermeture et la validation des modales, rendant le processus de création de catégories plus simple pour l'utilisateur.</p>
Création de plat	Gestion de l'ajout de plats dans les différentes catégories avec des informations détaillées	Utilisation de React Modal pour concevoir une fenêtre modale dédiée à la saisie des détails de chaque plat. La fenêtre doit comporter des champs pour le nom, le prix, la description et une option pour télécharger une photo pour chaque plat.	<p>1) React Modal propose une interface utilisateur dynamique et responsive pour la saisie des informations détaillées sans quitter la page.</p> <p>2) L'utilisation de React Modal permet une expérience de création de plats fluide et intuitive, augmentant ainsi la productivité des restaurateurs.</p>

Téléchargement d'images	Offrir la possibilité aux utilisateurs de charger des images	Intégration de Cloud Storage for Firebase pour faciliter le téléchargement d'images à partir d'ordinateurs ou d'appareils mobiles lors de la création de plats.	<p>1) Cloud Storage for Firebase offre un stockage cloud évolutif et sécurisé pour les fichiers.</p> <p>2) En utilisant cette solution, nous pouvons facilement intégrer la fonctionnalité d'upload d'image dans notre application et garantir la sécurité et la disponibilité des fichiers téléchargés.</p>
Accès aux menus précédents	Afficher les menus créés antérieurement.	Utiliser React Router pour la navigation.	<p>1)Rétrospective des menus : Offrir un accès visuel aux menus antérieurement créés pour une révision aisée.</p> <p>2)Gestion de navigation : Intégrer React Router pour faciliter la navigation entre les diverses sections de l'application, y compris la vue présentant les menus précédents.</p>

Affichage de la date de création des menus	Présenter la date de création des menus déjà existants.	Utiliser les données stockées dans la base de données.	<p>1)Contextualisation temporelle : Afficher la date de création des menus précédents pour une meilleure compréhension temporelle.</p> <p>2)Intégration de données : Utiliser les informations stockées dans la base de données pour extraire et présenter la date de création de chaque menu précédent de manière précise.</p>
Modification et suppression des menus précédents	Ajouter des fonctionnalités d'édition et de suppression.	Implémenter des fonctionnalités de modification et de suppression.	<p>1)Flexibilité d'action : Intégrer des fonctionnalités d'édition et de suppression pour une gestion aisée des menus.</p> <p>2)Accessibilité optimisée : Ajouter des boutons ou des icônes d'action pour permettre aux utilisateurs de modifier ou de supprimer les menus existants directement depuis la vue regroupant les menus précédents.</p>

Création d'un nouveau menu	Ajouter un bouton ou une fonctionnalité pour créer un nouveau menu	Fournir une méthode intuitive aux utilisateurs afin de créer un nouveau menu.	<p>1)Simplification du processus : Offrir une méthode intuitive pour créer un nouveau menu.</p> <p>2)Accessibilité accrue : Intégrer un bouton facilement repérable pour faciliter la création de nouveaux menus.</p>
Accès aux mentions légales	L'utilisateur doit etre en mesure de pouvoir en cliquant sur "mentions légales" avoir acces aux contenus des "mentions légales qui s'affiche dans une modale qu'il soit connecté ou non	React-modal	<p>1)Transparence améliorée : Accès facile aux mentions légales pour les utilisateurs.</p> <p>2)Expérience utilisateur fluide : Affichage des mentions légales dans une modale accessible.</p>
Accès aux tarifs de MenuMaker	L'utilisateur doit pouvoir cliquer sur l'onglet "Tarifs" et etre rediriger vers un nouvelle page.	Bouton de navigation	<p>1)Accessibilité facilitée : Bouton de navigation pour accéder rapidement aux tarifs.</p> <p>2)Navigation fluide : Utilisateurs redirigés vers la page des tarifs en un clic.</p>

Exportation vers Deliveroo	Integration et Connexion avec l'API Deliveroo	OAuth 2.0	<p>1)Intégration avec l'API Deliveroo via OAuth 2.0 pour permettre aux utilisateurs de MenuMaker de se connecter à leur compte Deliveroo.</p> <p>2)Cela autorise MenuMaker à accéder à leurs données et à effectuer des actions en leur nom.</p>
Partage sur Instagram	<p>Génération d'images au bon format .</p> <p>Integration et Connexion avec l'API Instagram</p>	<p>ImageMagick</p> <p>OAuth 2.0</p>	<p>1)L'utilisation de la bibliothèque ImageMagick pour créer des images au bon format à partir des images de menu existantes.</p> <p>2)Créer une connexion sécurisée avec Instagram, . Cela permet d'autoriser les utilisateurs à se connecter à leur compte Instagram et à autoriser votre application à accéder à leurs données ou à effectuer des actions comme le partage.</p>

Déconnexion	Gestion de l'authentification des utilisateurs via Firebase Auth	Utilisation de la fonction <code>signOut()</code>	<p>1) Sécurité renforcée : Déconnexion sécurisée via Firebase Auth pour protéger les données.</p> <p>2) Contrôle utilisateur : Utilisateurs peuvent se déconnecter à tout moment.</p>
Bouton de déconnexion	Disponibile sur toutes les pages .	Insertion dans un composant de navigation	<p>1) Accessibilité universelle : L'insertion du bouton de déconnexion dans un composant de navigation global assure sa disponibilité sur toutes les pages de l'application, offrant ainsi une expérience cohérente et pratique à l'utilisateur.</p> <p>2) Navigation simplifiée : Grâce à son emplacement dans le composant de navigation, le bouton de déconnexion est facilement accessible à tout moment, permettant aux utilisateurs de se déconnecter rapidement et en toute simplicité, quel que soit l'endroit où ils se trouvent dans l'application.</p>



Modification des informations utilisateur	Flexibilité dans la gestion des données	Stockage des informations utilisateur avec Firebase Firestore (NoSQL)	<p>1)Flexibilité améliorée : Le stockage des données utilisateur avec Firebase Firestore garantit une gestion souple et efficace des informations, offrant ainsi une expérience utilisateur optimisée.</p> <p>2)Simplification de la gestion : L'utilisation de Firebase Firestore simplifie la manipulation des données utilisateur, ce qui permet une mise à jour rapide et une adaptation aisée aux besoins changeants.</p>
Dashboard	Centralisation des fonctionnalités dans un tableau de bord intégré	Utilisation de React Router pour gérer la navigation	<p>1)Transition fluide : React Router assure une navigation sans accroc entre les fonctionnalités, améliorant ainsi l'expérience utilisateur.</p> <p>2)Organisation efficace : Le tableau de bord centralisé simplifie l'accès aux informations et facilite la gestion des fonctionnalités pour une productivité accrue.</p>

Branding du restaurateur	Gestion des attributs de marque (logo, palettes de couleurs)	Firebase Firestore	<p>1) Personnalisation avancée : En utilisant Firebase Firestore pour gérer les attributs de marque tels que le logo et les palettes de couleurs, les restaurateurs bénéficient d'un niveau élevé de personnalisation pour leur entreprise. Cela leur permet de créer une identité visuelle unique qui reflète leur marque et leur style.</p> <p>2) Centralisation des données Grâce à Firebase Firestore, toutes les informations de branding sont stockées de manière centralisée et sécurisée dans le cloud.</p>
--------------------------	--	--------------------	---

## 2 Liens avec le back-end

**Quel langage pour le serveur ?** Je suggère d'opter pour Node.js comme serveur en raison de son large usage, de sa capacité à s'adapter aux besoins croissants, et de sa compatibilité avec Firebase.

**Besoin d'une API ?** Effectivement, nous aurons besoin d'une API pour interagir avec Firebase afin de gérer l'authentification, le stockage des données, et d'autres fonctionnalités. Pour ce faire, nous utiliserons les API Firebase Authentication, Firebase Firestore et Firebase Storage.

En ce qui concerne la base de données, nous avons opté pour Firebase Firestore pour sa souplesse, sa capacité à s'adapter aux besoins évolutifs, et sa compatibilité native avec l'écosystème Firebase.

### **3 Préconisations concernant le domaine et l'hébergement**

**Nom du domaine : [menumaker.qwenta.com](https://menumaker.qwenta.com)**

**Justification :**

Cohérence de marque : Utiliser un sous-domaine de qwenta.com renforce l'identité de marque de Qwenta et montre que Menu Maker fait partie de leur offre de services.

Simplicité et mémorabilité : Un nom de domaine clair et concis est plus facile à retenir pour les utilisateurs.

SEO : Avoir un domaine lié à la marque principale peut améliorer le référencement sur les moteurs de recherche.

**Nom de l'hébergement:**

**Recommandation :**

**Nom de l'hébergement : Firebase Hosting**

**Justification :**

Intégration facile : Firebase Hosting s'intègre parfaitement avec Firebase Firestore et Firebase Authentication, qui sont déjà recommandés pour ce projet.

Performance : Firebase Hosting offre un CDN (Content Delivery Network) mondial, garantissant des temps de chargement rapides et une haute disponibilité.

Simplicité de déploiement : Firebase Hosting permet de déployer des applications web en quelques étapes simples, ce qui accélère le processus de mise en ligne.

Sécurité : Firebase Hosting gère automatiquement les certificats SSL, assurant que les connexions sont sécurisées.

Évolutivité : Firebase Hosting peut gérer une grande quantité de trafic, s'adaptant facilement aux besoins croissants de l'application sans nécessiter une gestion complexe de l'infrastructure.

**Adresse e-mail : [contact@menumaker.qwenta.com](mailto:contact@menumaker.qwenta.com)**

#### **4 Accessibilité**

- Les dernières versions de Chrome, Safari et Firefox.
- Le site devra être en version desktop. Pas de version mobile à développer ni à prévoir.
- L'application devra être accessible au minimum : navigable depuis le clavier, et lisible par un lecteur d'écran.

#### **5 Recommandations en termes de sécurité**

**Recommandation** : Utiliser Firebase Authentication pour la gestion des comptes utilisateurs.

- Authentification multi-facteurs (MFA) : Ajouter une couche supplémentaire de sécurité en activant l'authentification à deux facteurs (2FA).
- Hachage des mots de passe : Assurer que les mots de passe sont hachés et sécurisés avant d'être stockés.

#### **Justification** :

- Simplicité d'intégration : Firebase Authentication s'intègre facilement avec Firebase Firestore.

- Sécurité renforcée : La 2FA réduit le risque de compromission des comptes.

### **Plugins de sécurité:**

#### **Recommandation :**

- Utiliser des plugins de sécurité pour surveiller et protéger l'application, comme Helmet.js pour Express.js.

#### **Justification :**

- Protection contre les attaques courantes : Helmet.js aide à protéger l'application contre les attaques XSS, de clickjacking, etc.
- Meilleures pratiques : Utiliser des plugins de sécurité suit les meilleures pratiques en matière de développement sécurisé.

## **6 Maintenance du site et futures mises à jour**

### **Grandes lignes du contrat de maintenance:**

#### **Contrat de maintenance :**

##### **Support technique :**

Disponibilité : Assurer une disponibilité du support technique pendant les heures ouvrables.

Canaux de communication : Fournir un support via e-mail, téléphone et chat.

##### **Mises à jour régulières :**

Mises à jour de sécurité : Appliquer les patches de sécurité dès qu'ils sont disponibles pour protéger contre les vulnérabilités.

Mises à jour fonctionnelles : Ajouter de nouvelles fonctionnalités et améliorer les fonctionnalités existantes en fonction des retours des utilisateurs.

##### **Surveillance et monitoring :**

Surveillance continue : Utiliser des outils de monitoring pour surveiller la performance et la disponibilité du site.

Rapports mensuels : Fournir des rapports réguliers sur l'état du site, incluant les incidents et les mesures correctives prises.

**Sauvegardes régulières :**

- A. Fréquence : Effectuer des sauvegardes quotidiennes de la base de données et des fichiers importants.
- B. Restauration : Assurer un processus de restauration rapide en cas de perte de données.

**Gestion des incidents :**

Plan de réponse aux incidents : Mettre en place un plan de réponse rapide en cas de défaillance ou de cyberattaque.

Délai de réponse : Fixer des délais de réponse et de résolution en fonction de la gravité des incidents.