

PRÉSENTATION

Menu Maker by Qwenta

Sommaire

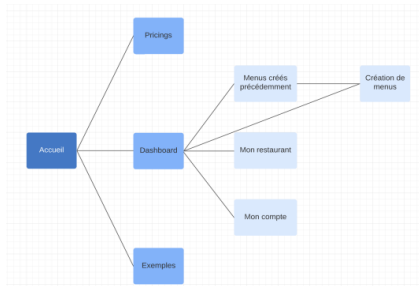
1. Contexte du projet
2. Aperçu de la maquette
3. Méthodologie utilisée
4. Tableau Kanban
5. Spécifications techniques
6. Veille technologique
7. Conclusion
8. Questions

Contexte du Projet

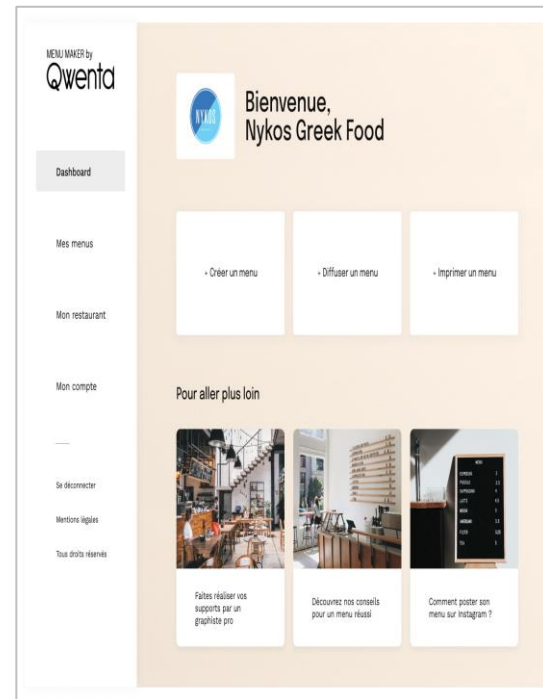
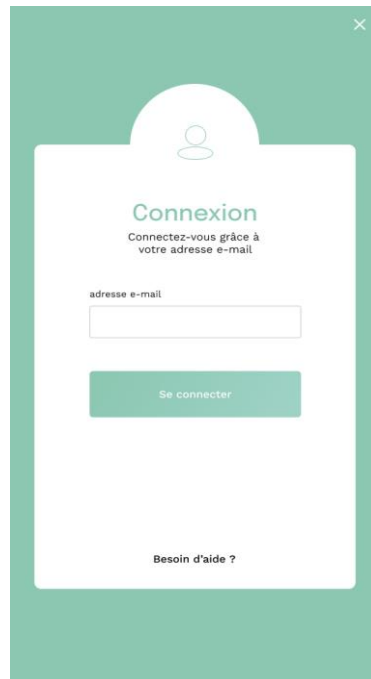
1 - Qu'est-ce que Menu Maker by Qwenta

Menu Maker by Qwenta est une application web destinée aux restaurateurs, leur permettant de créer et gérer des menus en ligne de manière personnalisée. Le projet vise à fournir une interface simple et réactive, avec des intégrations tierces pour la gestion des commandes via des plateformes comme Uber Eats et Deliveroo.

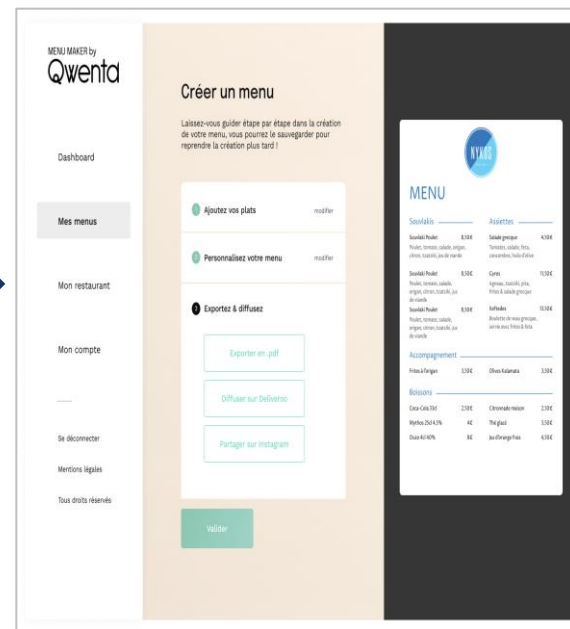
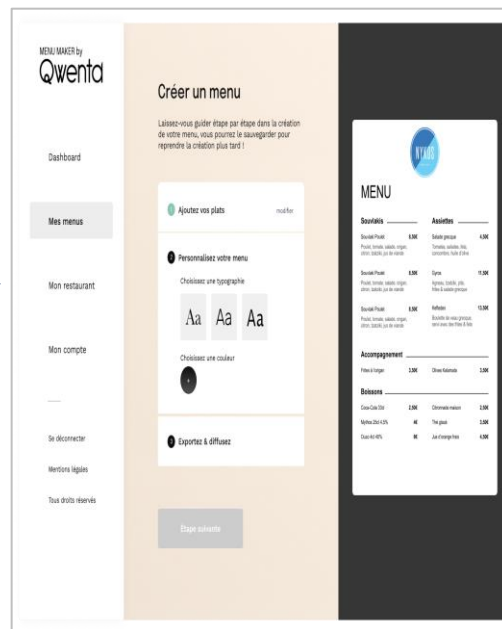
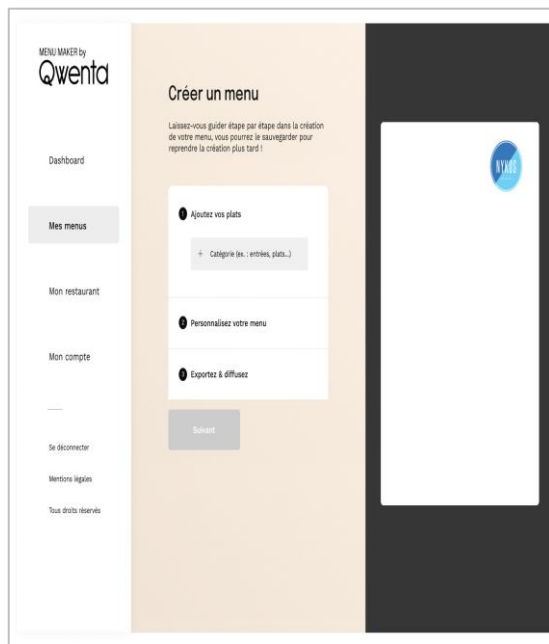
2 - Arborescence du site



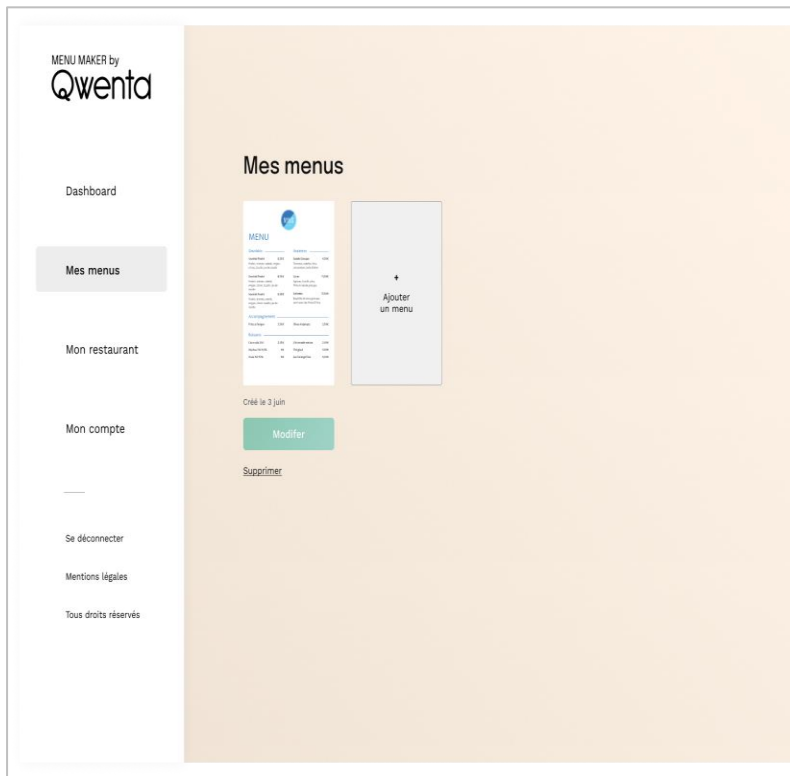
Aperçu de la maquette



Aperçu de la maquette



Aperçu de la maquette



La maquette de l'application montre une interface intuitive, où les restaurateurs peuvent :

- Créer des catégories de plats.
- Modifier leurs menus en temps réel.
- Gérer leurs tarifs et leur branding.

Fonctionnalités clés :

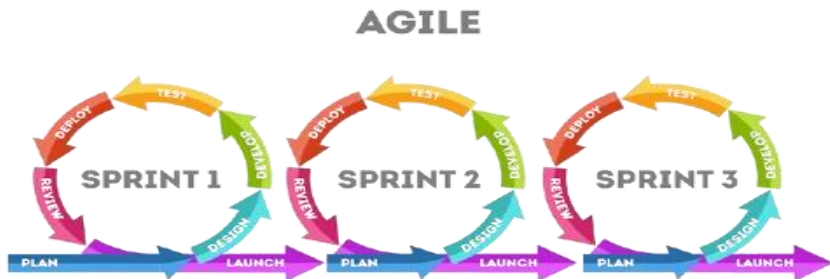
- Interface glisser-déposer pour la gestion des plats.
- Options de personnalisation visuelle (polices, couleurs).
- Visualisation des changements en temps réel.

Méthodologie utilisée

La méthode **Agile** a été adoptée pour ce projet, avec une structuration en **sprints** de deux semaines. Chaque sprint inclut la planification, le développement, et une rétrospective. Cette approche permet une flexibilité dans le développement et une adaptation rapide aux besoins évolutifs du client.

Avantages de cette méthode :

- Meilleure réactivité face aux ajustements demandés.
- Amélioration continue grâce aux feedbacks des parties prenantes.
- Suivi précis des tâches grâce au tableau **Kanban**.



Les approches Agiles

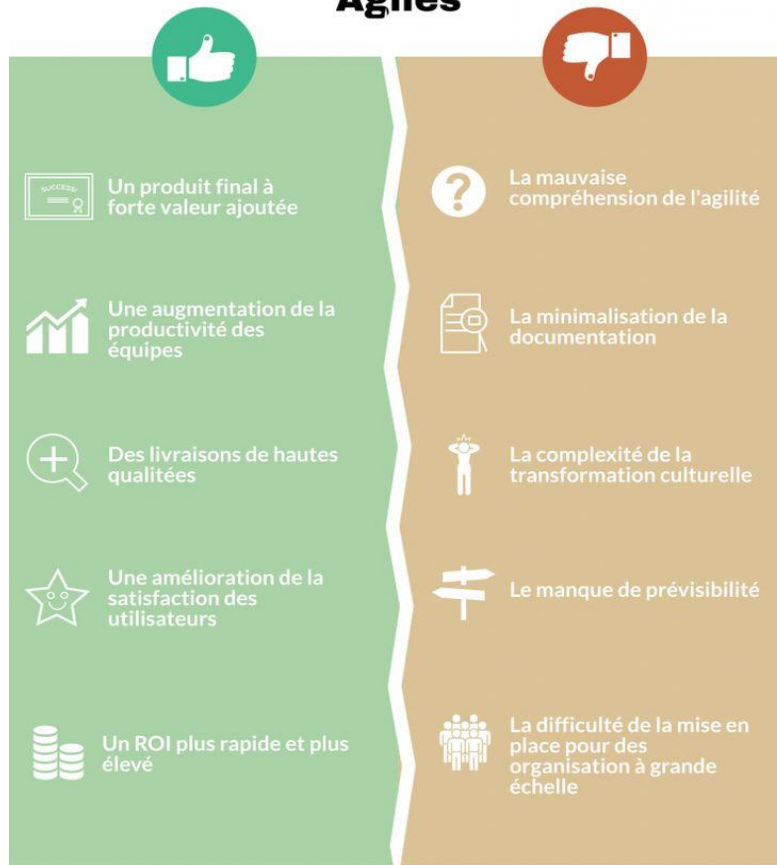


Tableau Kanban

Lien du tableau Kanban : <https://candle-makemake-ec7.notion.site/Offb7a8cab42800c9503f747d42b3f8e?v=b8a18b0db33f46098e0c9351bf97b6fb>

Aa Cartes	▼ Priorité	▼ Epic	☰ Product Owner	☀ État	☰ Equipe	# Story Point	+ ...
📄 Design maquette	P1 🔥	🍷 Design	Soufiane	● Terminé	UI/UX Designer	8	
📄 Landing non connectée	P1 🔥	🍷 Landing	Soufiane	● à faire	Frontend	3	
📄 Page login	P1 🔥	🔑 Connexion	Soufiane	● à faire	Frontend Backend	8	
📄 Catégorie de plat	P1 🔥	🍴 Création de menu	Soufiane	● à faire	Backend Frontend	8	
📄 Création de plat	P1 🔥	🍴 Création de menu	Soufiane	● à faire	Backend Frontend	8	
📄 Style de menu	P1 🔥	🍴 Création de menu	Soufiane	● à faire	Frontend	5	
📄 Exportation PDF	P1 🔥	🍴 Création de menu	Soufiane	● à faire	Backend Frontend	3	
📄 Commander l'impression d'un menu	P1 🔥	🍷 Back Office	Soufiane	● à faire	Backend Frontend	5	
📄 Menus précédents	P1 🔥	-	Soufiane	● à faire	Frontend Backend	5	
📄 Documentation des spécifications techn	P1 🔥	-	Soufiane	● En cours	Product owner Backend	8	
📄 Document Veille technologique	P1 🔥	-	Soufiane	● En cours	Frontend Product owner	5	
📄 Informations légales	P2	🍷 Landing	Soufiane	● à faire	Frontend	3	
📄 Tarifs	P2	🍷 Landing	Soufiane	● à faire	Frontend	3	
📄 Exportation Deliveroo	P2	🍴 Création de menu	Soufiane	● à faire	Frontend Backend	3	
📄 Partage sur Instagram	P2	🍴 Création de menu	Soufiane	● à faire	Backend Frontend	3	
📄 Déconnexion	P2	🔑 Connexion	Soufiane	● à faire	Frontend Backend	5	
📄 Infos utilisateur	P2	-	Soufiane	● à faire	Frontend Backend	3	
📄 Dashboard	P2	-	Soufiane	● à faire	Frontend Backend	13	
📄 Branding restaurateur	P3	-	Soufiane	● à faire	Frontend	3	
Total				● à faire		95	

Suivi du projet avec le Kanban

Modèle de carte

Chaque carte représente une **user story** et est assortie de labels indiquant l'**épïc** et les domaines de développement concernés. On peut attribuer chaque carte à un membre de l'équipe, avec un système de priorisation, comme ici avec un label rouge **P1** pour indiquer une priorité élevée.

La **description** inclut le texte de la user story, des informations sur les ressources nécessaires, des indications sur d'éventuelles extensions futures, ainsi que l'estimation de la **complexité** en **story points**.

Une **checklist** accompagne la carte pour suivre les critères d'acceptation, qui sont cochés une fois remplis. Les membres de l'équipe peuvent également commenter la carte pour ajouter des contributions ou des notes internes.

Enfin, il est conseillé de lier chaque carte à une branche GitHub spécifique pour un meilleur suivi du code lié à la tâche.

Page login

Type **Story**
 Priorité **P1**
 Product Owner Soufane
 Epic **Connexion**
 Equipe **Frontend** **Backend**
 # Story Point 8
 État **à faire**
 + Ajouter une propriété

Ajouter un commentaire...

Description

- En tant que restaurateur ayant un compte, je veux pouvoir me connecter.
- En tant qu'internaute, je veux pouvoir créer un compte automatiquement avec mon adresse mail.
- Gérer la connexion sécurisée des utilisateurs à leur compte via un formulaire de connexion. Une fois connectés, ils pourront accéder à leur espace de gestion des menus et commandes.

Spécifications techniques

- Solution technologique** : Utilisation de [Firebase Authentication](#) pour la gestion sécurisée de la connexion des utilisateurs. L'interface est développée en React.js avec une gestion de session intégrée.
- Liens avec le back-end** : Utilisation d'API [Firebase](#) pour l'authentification. Une fois connecté, l'utilisateur reçoit un [token Firebase](#) pour gérer sa session.
- Sécurité** : Utilisation de [Firebase Authentication](#) avec des options comme la vérification par e-mail et la gestion de session sécurisée.
- Recommandation en termes de sécurité** : HTTPS activé pour toutes les connexions et protection contre les attaques de force brute via les fonctionnalités de [Firebase](#).

Succès

- ☐ Cette fenêtre s'ouvre sous forme de modale
- ☐ L'utilisateur peut entrer son adresse mail
- ☐ Qu'il se soit déjà connecté ou non, un mail lui est envoyé pour lui permettre de s'authentifier, ou au contraire de confirmer son mail pour accéder pour la première fois à l'application
- ☐ Un lien "Besoin d'aide" permet d'envoyer directement un e-mail à nos équipes

Design

[Lien Maquette Figma](#)

Planning Poker

Pour évaluer la complexité des 95 user stories, l'équipe utilisera la méthode **Planning Poker**. Cette technique permet d'estimer de manière collaborative les efforts nécessaires pour accomplir chaque tâche, en tenant compte de la perception collective de la difficulté.

Le **Planning Poker** fonctionne comme suit :

- **Présentation de la user story** : Le product owner ou le responsable de projet présente la user story à l'équipe, en expliquant les objectifs et les critères d'acceptation.
- **Discussion** : L'équipe discute brièvement de la tâche pour s'assurer que tout le monde a une compréhension commune de la user story.
- **Vote simultané** : Chaque membre de l'équipe choisit une carte représentant son estimation de la complexité de la tâche (1, 2, 3, 5, 8, 13, etc. selon la suite de Fibonacci). Tous révèlent leur carte en même temps pour éviter l'influence des uns sur les autres.
- **Consensus** : Si les estimations varient, une discussion a lieu pour comprendre les écarts et ajuster l'estimation finale en fonction du consensus.



Méthode MOSCOW

Pour établir la priorité des fonctionnalités, la méthode **MoSCoW** est utilisée. Celle-ci classe les tâches en fonction de leur importance :



Essentielles (Must Have)	Importantes (Should Have)	Secondaires (Could Have)	À exclure (Won't Have)
Fonctionnalités critiques pour le fonctionnement de l'application, telles que la création des menus, la gestion des utilisateurs via Firebase Authentication, et l'exportation des données en PDF.	Fonctionnalités nécessaires mais non bloquantes, comme l'intégration avec les APIs Instagram et Deliveroo .	Fonctionnalités ajoutant de la valeur, mais non indispensables au lancement, telles que l'ajout d'animations dans l'interface utilisateur.	Fonctionnalités qui ne seront pas incluses dans cette version du projet, telles qu'une version mobile complète.

Technologies

- FRONT-END :

- **React.js** : Bibliothèque JavaScript utilisée pour construire l'interface utilisateur. Permet la création de composants réutilisables et une gestion efficace de l'état de l'application.
- **Axios** : Librairie HTTP utilisée pour les requêtes API, facilitant la communication entre le front-end et le back-end.
- **SCSS** : Préprocesseur CSS utilisé pour styliser l'application avec des fonctionnalités avancées comme les variables, les mixins et les imbrications, rendant le code plus modulable.

- BACK-END :

- **Node.js** : Environnement JavaScript côté serveur, utilisé pour créer l'API back-end, gérer la logique métier et assurer la communication entre Firebase et l'application front-end.
- **Firebase** : Plateforme complète utilisée pour gérer l'authentification des utilisateurs, la base de données en temps réel (Firestore) et l'hébergement sécurisé de l'application.

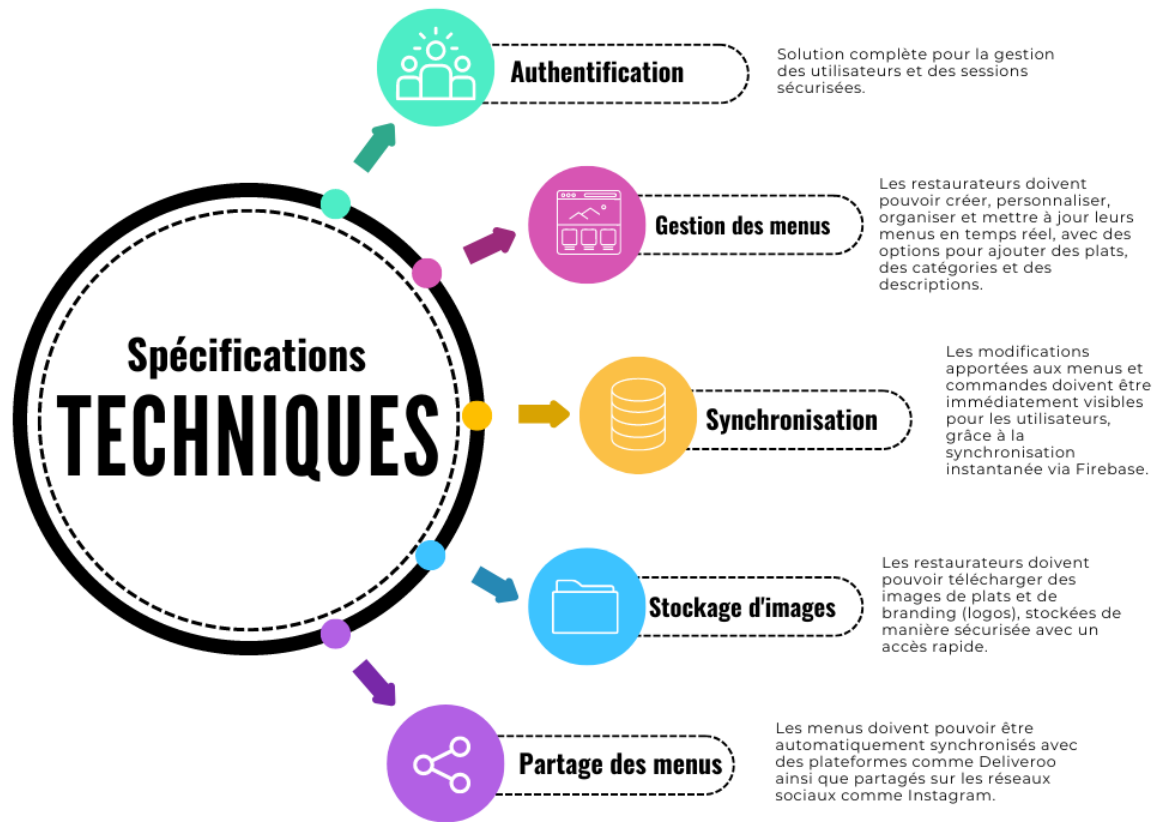
FRONT-END



BACK-END



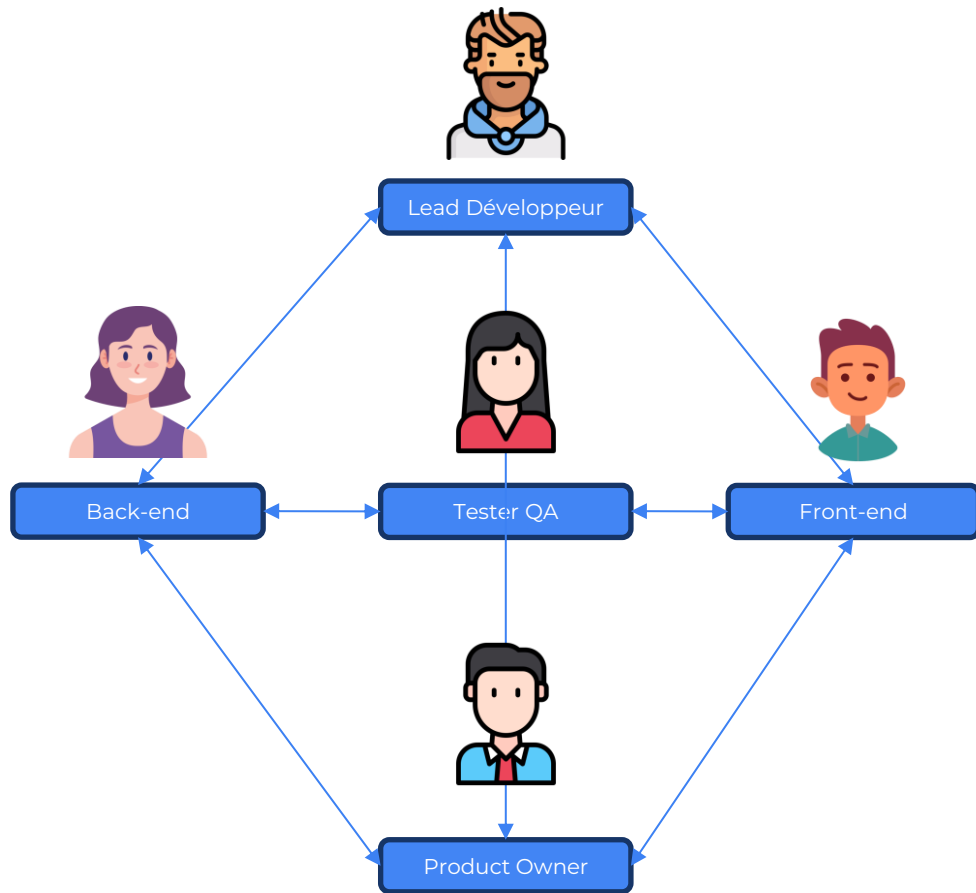
Spécifications techniques



Effectif de l'équipe

L'équipe en charge du développement du projet **Menu Maker by Qwenta** se compose de plusieurs membres aux compétences complémentaires :

- **Développeur Front-End** : En charge de l'interface utilisateur, utilisant **React.js** et **SCSS** pour assurer une expérience fluide, réactive, et visuellement cohérente avec les besoins du projet.
- **Développeur Back-End** : Responsable de l'architecture serveur, des APIs et de la gestion des données en temps réel via **Firebase** et **Node.js**. Il assure également l'intégration avec des plateformes externes comme Deliveroo.
- **Lead Développeur** : Responsable de la supervision technique globale, il veille à la bonne coordination entre le front-end et le back-end, prend les décisions techniques majeures et s'assure de la qualité du code.
- **Testeur QA (partiel)** : Chargé d'assurer la qualité des fonctionnalités à travers des tests unitaires, fonctionnels, et d'intégration.
- **Product Owner** : Coordonne l'équipe, priorise les tâches, et fait le lien avec les parties prenantes.



Authentification

Voici un **flowchart** détaillant les étapes et le fonctionnement de l'authentification avec Firebase, node et react :

Début : Saisie des identifiants (Ovale)

L'utilisateur entre son **email** et **mot de passe** dans un formulaire React.js.

Vérification des identifiants dans Firebase (losange pour la décision)

Envoi des informations à **Firebase Authentication** pour vérification.

"Identifiants corrects ?" (flèche pour Oui/Non)

- **Oui** : Passe à l'étape de succès.
- **Non** : Message d'erreur et retour à la saisie des identifiants.

Succès : Génération du token (rectangle)

Si les identifiants sont corrects, Firebase génère un **token JWT**.

Échec : Message d'erreur (rectangle)

Si les identifiants sont incorrects, affichage d'un message d'erreur (ex. "Mot de passe incorrect").

Stockage du token JWT (rectangle)

Le **token JWT** est stocké dans le navigateur (localStorage ou cookie sécurisé).

Vérification du token à chaque requête (losange pour la décision)

À chaque requête, le token est envoyé pour valider la session.

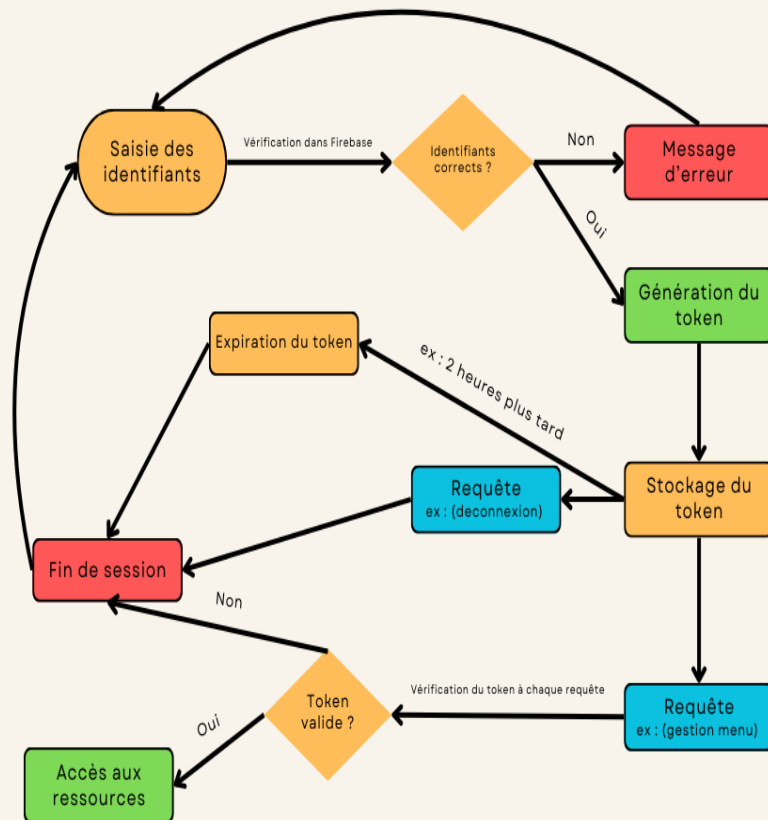
- **Oui** : Si le token est valide, passe à l'étape suivante.
- **Non** : Si le token est invalide, passe à la fin de session.

Accès aux ressources (rectangle)

Si le token est valide, l'utilisateur peut accéder aux ressources protégées comme la gestion des menus.

Fin de session (déconnexion ou expiration du token) (rectangle)

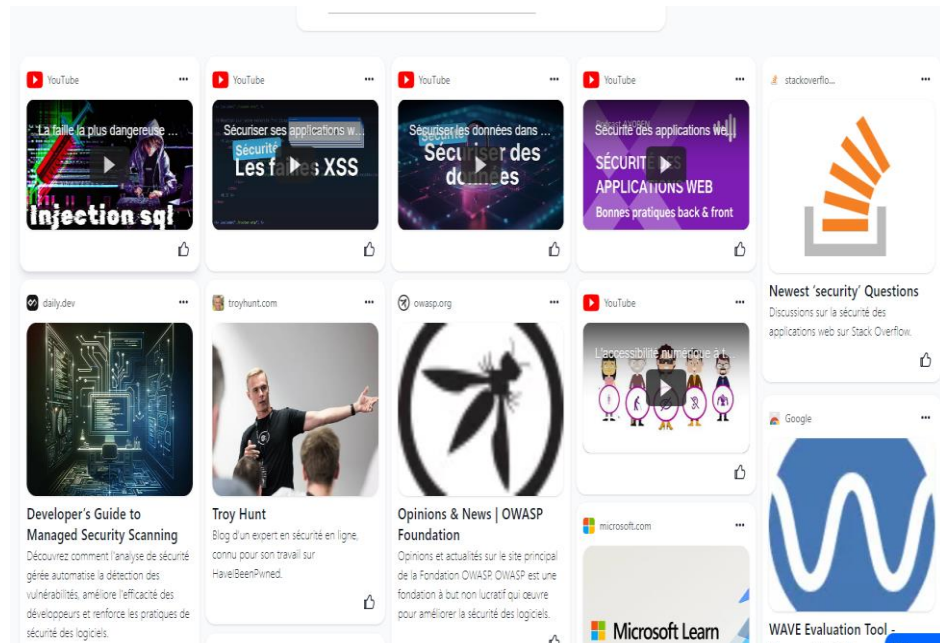
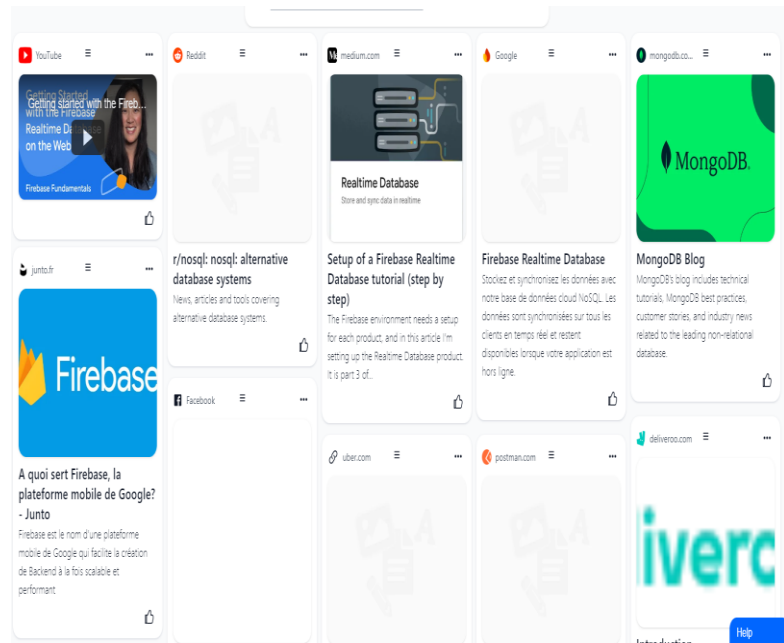
Lorsque le token expire ou que l'utilisateur se déconnecte, la session prend fin et le token est supprimé.



Veille Technologique

La veille technologique est une phase cruciale dans la planification du projet Menu Maker by Qwenta. Elle permet d'identifier les technologies les plus adaptées à l'application à développer et de rester informé des tendances actuelles du développement web.

[Lien Wakelet](#)



Veille Technologique

Méthode de classification des sources d'information

Les sources d'information sont classées en fonction de trois critères principaux :

- **Autorité** : Seules les sources crédibles et bien établies dans le domaine du développement sont utilisées, comme les sites officiels (Google, Mozilla) ou les blogs reconnus (CSS-Tricks, Dev.to).
- **Fraîcheur** : Les articles les plus récents sont privilégiés, surtout pour des sujets qui évoluent rapidement comme les frameworks (React, Node.js) ou la sécurité.
- **Pertinence** : Les sources doivent répondre directement aux besoins techniques du projet (front-end, back-end, API).

Choix des sources :

- **Axe technique (React.js)** : **daily.dev** est choisi car il agrège les articles de sources multiples sur des sujets techniques, comme les meilleures pratiques en **React.js**. Grâce à cela, l'équipe peut facilement identifier les bonnes pratiques de développement front-end, y compris l'utilisation des hooks et la gestion des états, qui sont essentiels pour l'interface utilisateur réactive.
- **Axe général (développement web et sécurité)** : **Reddit (r/webdev)** est une plateforme qui offre des discussions sur des problématiques concrètes et variées en développement. Par exemple, des échanges sur les pratiques de sécurité dans l'utilisation des **APIs** ont permis de définir des règles de sécurité solides pour l'intégration avec Firebase.

Veille Technologique

La veille technologique a joué un rôle central dans l'élaboration des spécifications techniques en garantissant que les solutions choisies sont à jour, performantes et en accord avec les standards actuels. Par exemple :

- **React.js** a été sélectionné pour le front-end après l'analyse de plusieurs articles sur ses performances et sa modularité, permettant de garantir une interface fluide et réactive.
- Pour le **back-end**, la veille a permis de choisir **Firebase** comme solution d'hébergement et de base de données, pour sa scalabilité et son intégration facile avec des outils comme **Firebase Authentication** et **Firebase Realtime Database**.
- L'intégration de l'API **Deliveroo** a été décidée après des discussions sur Reddit et Stack Overflow, où la communauté partage des exemples de cas réels et des solutions pratiques pour la gestion d'APIs.

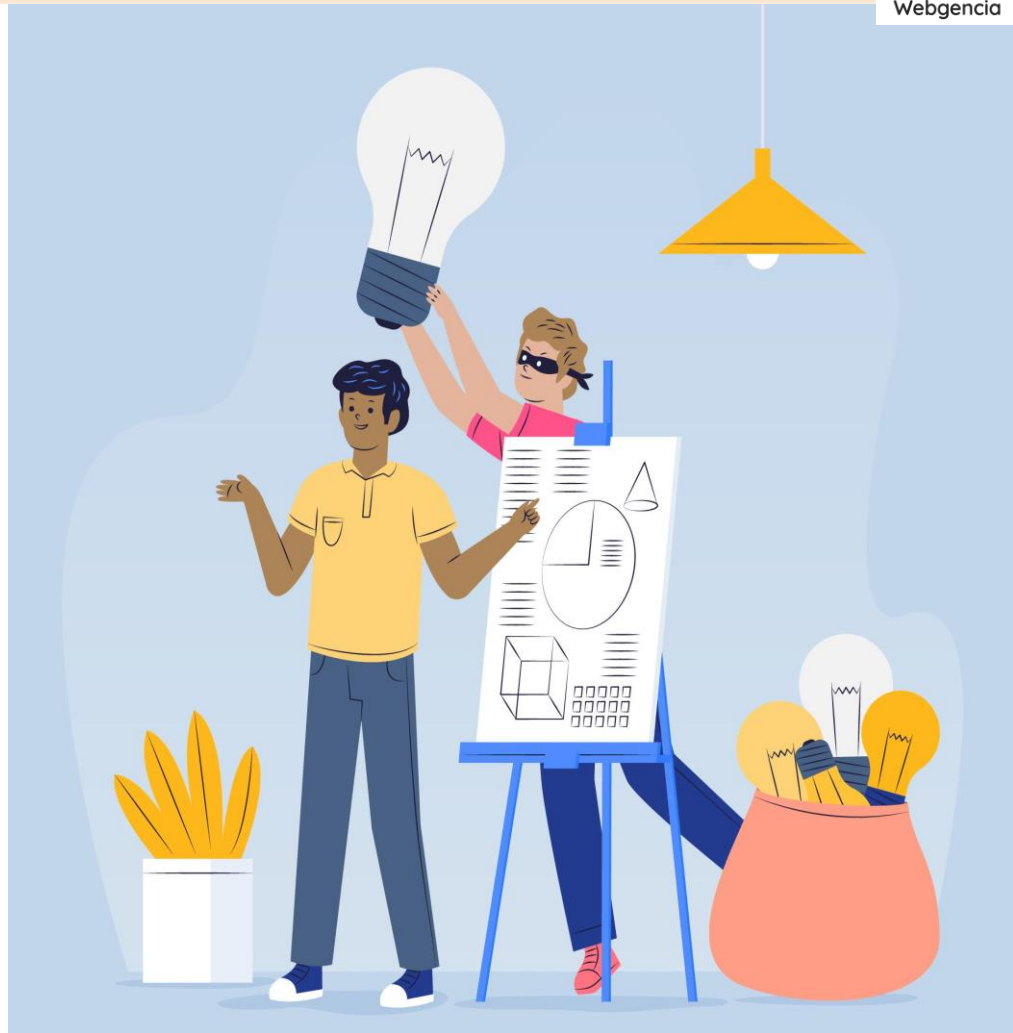
Cette veille a permis d'optimiser les choix techniques, garantissant ainsi une application robuste et évolutive, tout en respectant les besoins de performance et de sécurité identifiés dans les spécifications.



BLOGGING

Conclusion

- Le projet **Menu Maker by Qwenta** a été soigneusement élaboré en combinant une veille technologique rigoureuse, des choix techniques modernes et une planification détaillée. Grâce à l'utilisation de **React.js** pour le front-end, **Firebase** pour la gestion des données et l'authentification, et l'intégration des **APIs externes** telles que Instagram et Deliveroo..., l'application est conçue pour être **scalable, sécurisée** et **facile à maintenir**.
- Les spécifications techniques ont été définies avec précision, couvrant l'authentification sécurisée, la gestion des menus en temps réel, et l'exportation de rapports. Un processus de **tests unitaires** a également été mis en place pour garantir la qualité du code.
- Avec une équipe dédiée composée d'un **lead développeur**, d'un **développeur back-end** et d'un **développeur front-end**, le projet assure une coordination efficace et des mises à jour régulières pour répondre aux évolutions du marché.



QUESTIONS ?