

Sommaire

- 1. Sommaire
- 2. Définitions des spécifications techniques
- 3. Étapes par étapes du projet
- 4. Choix technologique
- 5. Les besoins
- 6. Les contraintes
- 7. Solution, description et justification
- 8. Liens avec le back-end
- 9. Domaine d'hébergement
- 10. Accessibilité
- 11. Services tiers
- 12. Sécurité
- 13. Maintenance du site et futur mise à jour



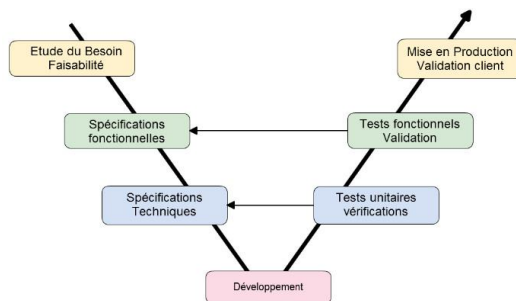
2. Définitions des spécifications techniques

Les spécifications techniques, c'est quoi ?

La spécification fonctionnelle et technique est un document constitué d'un ensemble d'exigences relatives à la mise en œuvre du projet

Il contient, entre autres, des informations sur:

- les cadres de programmation
- la structure de produit planifiée et les fonctionnalités nécessaires, qui formeront le projet final et répondront aux attentes commerciales du client.



3. Étapes par étapes du projet

1. Fixer des objectifs:

- Définir l'idée de départ (qui a eu l'idée du projet et pourquoi).
- Définir une vision du projet.
- Définir les caractéristiques de base du projet qui constitue le point de départ.
- Définir les besoins de chacun lors de la conception (utilisateur connecté ou non connecté).

2. Spécifiez les futurs utilisateurs:

- Définir qui sera l'utilisateur du site web ou de l'application.
- Définir l'âge et le sexe.
- Définir les besoins

3. Exemples de visuel.

- Présentez des produits similaires au projet

La présentation d'exemples permet une vision de départ pour la suite des travaux.

4. Planifiez une structure

Au cours du projet, la structure peut évoluer.

Une planification est une procédure extrêmement précieuse, créer un plan ou même un prototype permet de se retrouver.

5. Liste des besoins

Spécifiez les besoins requis pour l'environnement dans lequel le projet doit fonctionner, les infrastructures, la connexion et la sécurité des utilisateurs.

4. Choix technologique



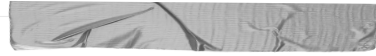
Pourquoi utilisé React et Express js ?

React est très simple à lire et Javascript simple à écrire

Express est une infrastructure d'application (framework), écrit en JavaScript et hébergée dans l'environnement d'exécution node.js.

Il s'agit d'un framework minimaliste qui permet de développer rapidement des applications.

5. Les besoins



Que pourra faire l'utilisateur ?

- Créer un menu
- Personnaliser un menu
- Diffuser un menu
- Imprimer un menu

6. Les contraintes

L'ajout d'une catégorie doit pouvoir se faire directement sur l'écran de création de menu depuis une modale

7. Solution, description et justification

8. Liens avec le back-end

Quel langage utilisé pour le serveur?.NodeJs

À propos de Node.js

En tant qu'environnement d'exécution JavaScript asynchrone piloté par les événements, Node.js est conçu pour générer applications réseau évolutives.

Node.js et multi-plateforme permet de créer des applications rapide et évolutives.

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

Pourquoi utiliser Node JS

1. Node.js est un excellent choix pour les API Rest

API : Les Application Programming Interfaces. Les API Rest sont utilisés pour les échanges de données entre serveurs et clients.

L'un des avantages de la technologie Node.js est la possibilité d'exécuter plusieurs requêtes vers le serveur simultanément, car c'est un système « single thread »

non bloquant. donc Node.js est capable de gérer une multitude de requêtes sans attendre que la requête lancée précédemment soit terminée.

2. Excellentes performances

Avec l'avantage de traiter une multitude d'opérations dans sa boucle d'événements, le Node.js offre des performances incroyables.

De plus, Node.js est léger et la technologie est basée sur le très puissant moteur V8 de Google Chrome, ce qui offre des

performances incroyables aux applications web

qui doivent chercher des informations et exécuter une multitude de requêtes.

3. Beaucoup de flexibilité

Un grand écosystème de bibliothèques open-source offre la possibilité aux développeurs d'ajouter les modules nécessaires pour une application web Node.js.

Un développeur qui maîtrise le JavaScript peut concevoir le backend et le frontend d'une application web. La très grande communauté JavaScript

(le langage de programmation le plus utilisé sur le web) travaille constamment sur la création de nouveaux modules. Des entreprises telles que Netflix, PayPal et IBM utilisent la technologie Node.js.

4. Une bonne stack technique

En utilisant du JavaScript côté serveur en backend et côté interface frontend avec les différents frameworks frontend tels que le

React.js, le Next.js, Vue.js ou Angular.js,

votre application web utilise du JavaScript partout.

9. Domaine d'hébergement

Préconisations concernant le domaine et l'hébergement

Le nom du domaine sera probablement un sous domaine de Qwenta.

10. Accessibilité du site

Accessibilité

L'application devra être accessible au minimum : navigable depuis le clavier, et lisible par un lecteur d'écran.

Compatibilité navigateur.

Pour le moment, on se contente de la compatibilité avec les dernières versions de Chrome, Safari et Firefox
Le site devra être en version desktop. Pas de version mobile à développer ni à prévoir.



11. Services tiers

Services tiers

Noms et justification de ceux qui devront être implémentés pour le bon fonctionnement et le suivi du site

Recommandations en termes de sécurité

Accès aux comptes, plugins...

Maintenance du site et futures mises à jour

Grandes lignes du contrat de maintenance.

12. Sécurité

Recommandations de sécurité

- Mot de passe 12 caractères minimum
- Utilisations de HTTPS
- Authentification à 2 facteurs
- Limitation de connexion
- Gestion des cookies
- Expirations des sessions
- Surveillance du domaine



13. Maintenance du site et futur mise a jour

Maintenance

- Mise à jour régulières
- Sauvegarde régulières
- Gestion des erreurs
- Optimisations des performances
- Compatibilité multiplateforme
- Amélioration de l'expérience utilisateur "questionnaire"
- Test de fonctionnalité
- Suivi des statistique

