

Cahier des Charges

Application Météorologique Web/Mobile

Sanlove Eden ALEXIMA

October 19, 2024

Contents

1	Contexte et objectif du projet	2
1.1	Présentation du projet	2
1.2	Objectifs principaux	2
2	Public cible	2
2.1	Définir les utilisateurs	2
3	Fonctionnalités attendues	2
3.1	Fonctionnalités principales pour l'utilisateur	2
3.2	Fonctionnalités avancées	3
3.3	Fonctionnalités administratives	3
4	Technologies utilisées	3
4.1	Frontend (côté utilisateur)	3
4.2	Backend (côté serveur)	3
4.3	Infrastructure	3
5	Interface utilisateur (UI/UX)	4
6	Contraintes techniques et fonctionnelles	4
7	Planification du projet	4
7.1	Phases du projet	4
7.2	Délais et Budget	4
8	Critères de validation	5
9	Maintenance et évolutions futures	5
9.1	Suivi	5
9.2	Évolutions	5

1 Contexte et objectif du projet

1.1 Présentation du projet

Cette application a pour but de fournir des données météorologiques en temps réel ainsi que des prévisions à des utilisateurs pour des usages variés (Urgeo, agriculture, tourisme, navigation, etc.). Elle permettra également de consulter des données historiques et d'accéder à des alertes météorologiques importantes. Les utilisateurs doivent pouvoir accéder à ces informations de manière simple et intuitive. Une application hybride accessible via un navigateur web et une application mobile (Android/iOS).

1.2 Objectifs principaux

Les principaux objectifs de l'application sont :

- Afficher les prévisions météo à court et long terme.
- Offrir un accès aux données météorologiques historiques.
- Proposer des cartes interactives pour visualiser les prévisions.
- Fournir des notifications en temps réel pour les alertes météo.

2 Public cible

2.1 Définir les utilisateurs

Les utilisateurs ciblés par cette application sont :

- **Utilisateurs professionnels** :météorologues,organisations spécialisées dans le climat, agriculteurs, pilotes, capitaines de navires, etc.
- **Utilisateurs grand public** : voyageurs, passionnés de météo, etc.
- **Administrateurs** :Urgeo, gestionnaires des données et des utilisateurs.

3 Fonctionnalités attendues

3.1 Fonctionnalités principales pour l'utilisateur

- Accès aux prévisions météo par localisation géographique (ville, région, pays).
- Consultation des prévisions horaires et journalières.
- Visualisation sur une carte (intégration avec Google Maps ou autre).
- Affichage des données météo en temps réel (température, précipitations, humidité, vent, etc.).
- Consultation des données historiques pour une période donnée.
- Notifications push pour les alertes météo (tempêtes, orages, etc.).

3.2 Fonctionnalités avancées

- Personnalisation des alertes (types d'alertes, régions spécifiques).
- Utilisation de graphiques pour visualiser les tendances météo.
- Partage des prévisions sur les réseaux sociaux.
- Partage des prévisions sur les réseaux sociaux.

3.3 Fonctionnalités administratives

- Gestion des utilisateurs et des permissions.
 - Interface de gestion des Capteurs
 - * Ajout de nouveaux capteurs
 - * Surveillance des capteurs
 - * Configuration des capteurs
 - * Type de données collectées sur le capteur
 - * Configuration des seuils d'alerte
 - * Indicateurs de statut pour chaque capteur
 - * Historique des données / Variations
 - * Alerte en temps réel(email, notifications, SMS)
 - * Journal des erreurs pour surveiller les dysfonctionnements ou les pannes
 - * Tableau de bord de tous les capteurs

4 Technologies utilisées

4.1 Frontend (côté utilisateur)

- Web : Ionic & Angular.
- Mobile : Ionic et Angulars.
- Langage : JavaScript.

4.2 Backend (côté serveur)

- Langage : C#.
- Base de données : MongoDB.
- API : ASP .NET Core.

4.3 Infrastructure

- Hébergement sur le cloud (AWS).
- Notifications push via Firebase Cloud Messaging.
- Versionnage(Github).

5 Interface utilisateur (UI/UX)

- Design simple et intuitif.
- Accès facile aux informations essentielles (température, précipitations, etc.).
- Couleurs adaptées à la météo (ex. : bleu pour le ciel, gris pour les nuages, etc.).
- Responsive design pour s'adapter aux écrans mobiles, tablettes, et ordinateurs.

6 Contraintes techniques et fonctionnelles

- Compatibilité avec iOS, Android, et les principaux navigateurs web.
- Sécurité des données utilisateurs (ex. : localisation) et confidentialité des informations personnelles.
- Performance : l'application doit être rapide et fluide, même avec une connexion faible.
- Connectivite: Etude pour une meilleure reseau de connectivite des capteurs avec l'API(WIFI/Bluetooth/Zigbee/Z-wave/Cellulaire).
- Protocole: Etude du protocole de communication(HTTP/HTTPS/MQTT/Websocket).
- Configuration des capteurs

7 Planification du projet

7.1 Phases du projet

1. Analyse des besoins et spécifications détaillées.
2. Développement du prototype (wireframes et maquettes).
3. Etude des capteur
4. Développement frontend et backend.
5. Intégration des API météorologiques.
6. Tests (unitaires, fonctionnels, utilisateurs).
7. Mise en production et maintenance.

7.2 Délais et Budget

- Estimation délai: 8 mois
- Budget nécessaire selon les ressources (développeurs, serveurs, outils).

8 Critères de validation

- Validation des fonctionnalités par des tests utilisateurs.
- Conformité aux exigences et fonctionnalités définies.
- Performance et fiabilité (temps de réponse, stabilité, etc.).

9 Maintenance et évolutions futures

9.1 Suivi

La maintenance de l'application comprendra des mises à jour régulières pour corriger les bugs et assurer sa pérennité.

9.2 Évolutions

De nouvelles fonctionnalités pourront être ajoutées, comme l'intégration de nouvelles données météo ou des prévisions saisonnières.