Projet de réalisation d'une Application Météorologique Web/Mobile

Libellé

Remis à

Sanlove Eden ALEXIMA
Jackyno JEAN-BAPTISTE
Philipe JOSE

October 22, 2024

Contents

1 Contexte et objectif du projet **3**
1.1 Présentation du projet 3
1.2 Objectifs principaux 3

2 Public cible **3**
2.1 Définir les utilisateurs 3

3 Fonctionnalités attendues **3**
3.1 Fonctionnalités principales pour l'utilisateur 3
3.2 Fonctionnalités avancées 4
3.3 Fonctionnalités administratives 4

4 Fonctionnalités envisagées **4**
4.1 Fonctionnalité optionnelle : Contrôle à distance d'un climatiseur 4

5 Technologies utilisées **6**
5.1 Frontend (côté utilisateur) 6
5.2 Backend (côté serveur) 6
5.3 Infrastructure 6
5.4 Autres outils 6

6 Interface utilisateur (UI/UX) **6**

7 Contraintes techniques et fonctionnelles **6**

8 Planification du projet **7**
8.1 Phases du projet 7
8.2 Délais et Budget 7

9 Critères de validation **7**

10 Maintenance et évolutions futures **7**
10.1 Suivi 7
10.2 Évolutions 7

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 . 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** : gestionnaires des données et des utilisateurs(UrGEO)

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.
- Utiliser des API de cartes telles que "Leaflet", "Google Maps" ou "Mapbox" pour afficher des cartes interactives montrant les stations météorologiques et leurs relevés.
- 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.).

- Intégrer des annotations ou des événements clés (ex. une tempête ou un pic de température) sur les graphiques pour un meilleur suivi.
- Ajouter des couches de cartes spécifiques (par ex. affichage de la vitesse du vent, des zones de précipitations, ou des températures par zones colorées).
- Permettre un filtrage basé sur des critères spécifiques (afficher uniquement les zones avec des températures supérieures à 30°C, par exemple).

3.2 Fonctionnalités avancées

- Personnalisation des alertes (types d'alertes, régions spécifiques).
- Personnalisation des données météorologique
- Utilisation de graphiques pour visualiser les tendances météo.
- 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)
  - Jourrnal des erreurs pour surveiller les disfonctionnements ou les pannes
  - Tableau de bord de tous les capteurs

4 Fonctionnalités envisagées

4.1 Fonctionnalité optionnelle : Contrôle à distance d'un climatiseur

Description :
L'application pourrait intégrer une fonctionnalité de contrôle à distance d'un climatiseur, déclenchée automatiquement lorsque la température enregistrée par les capteurs dépasse un certain seuil (défini par l'utilisateur ou configuré par défaut). Cette fonctionnalité viserait à maintenir une température ambiante agréable sans intervention manuelle.

Objectif :

- Garantir le confort de l'utilisateur en ajustant automatiquement la température de l'environnement via un dispositif de climatisation.

- Contribuer à la gestion énergétique en fonction des conditions climatiques mesurées.

Fonctionnement prévu :

1. **Détection de la température :** Les capteurs mesurent en temps réel la température ambiante et envoient les données à l'application.
2. **Analyse des données :** Lorsque la température dépasse un certain seuil (à définir par l'utilisateur), une alerte est générée.
3. **Action de contrôle :** L'application enverrait un signal de contrôle au climatiseur (via un protocole approprié comme IR, Wi-Fi, ou API spécifique), demandant une mise en marche ou une modification des paramètres (comme baisser la température).
4. **Retour d'état :** Le climatiseur renverrait un retour d'état (si compatible) pour indiquer que l'action a été exécutée.

Pré-requis techniques :

- Un climatiseur compatible avec une interface de commande à distance (ex : via infrarouge, Wi-Fi, ou autre protocole supporté).
- Une API ou un protocole de communication permettant l'intégration avec l'application.

Étude de faisabilité :
Cette fonctionnalité nécessite une *étude de faisabilité* qui sera menée pendant le projet. Les aspects suivants devront être étudiés :

- **Compatibilité matérielle :** Identification des climatiseurs compatibles (par exemple, modèles équipés de Wi-Fi, d'IR, ou d'une API).
- **Protocole de communication :** Vérification de l'existence de solutions logicielles et matérielles pour permettre le contrôle à distance (via infrarouge, Wi-Fi, etc.).
- **Limites techniques :** Analyse des éventuelles limitations liées aux modèles de climatiseurs disponibles et à leur capacité à être commandés à distance.

Évaluation des risques :

- **Incompatibilité des climatiseurs :** Certains modèles de climatiseurs ne peuvent pas être contrôlés à distance via une API standard.
- **Complexité de l'intégration :** L'intégration pourrait nécessiter des protocoles spécifiques ou des équipements supplémentaires (comme des récepteurs infrarouges).
- **Coût supplémentaire :** L'achat de matériels spécifiques pour la communication ou la commande des climatiseurs pourrait entraîner des coûts supplémentaires.

Décision de mise en œuvre :
En fonction des résultats de l'étude de faisabilité, cette fonctionnalité sera soit :

- **Implémentée** si elle est jugée techniquement et économiquement viable,
- **Abandonnée ou reportée** dans le cas contraire.

5 Technologies utilisées

5.1 Frontend (côté utilisateur)

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

5.2 Backend (côté serveur)

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

5.3 Infrastructure

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

5.4 Autres outils

- Versionnage(Github).
- Suivi et evaluation(JIRA)
- Test endpoint(Postman)

6 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.

7 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.
- Connectivité : Etude pour une meilleure reseau de connectivité des capteurs avec l'API (WIFI/Bluetooth/Zigbee/Z-wave/Cellulaire).

- Protocole: Etude du protocole de communication adequat (HTTP/HTTPS/MQT-T/Websocket) .
- Configuration des capteurs

8 Planification du projet

8.1 Phases du projet

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

8.2 Délais et Budget

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

9 Critères de validation

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

10 Maintenance et évolutions futures

10.1 Suivi

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

10.2 Évolutions

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