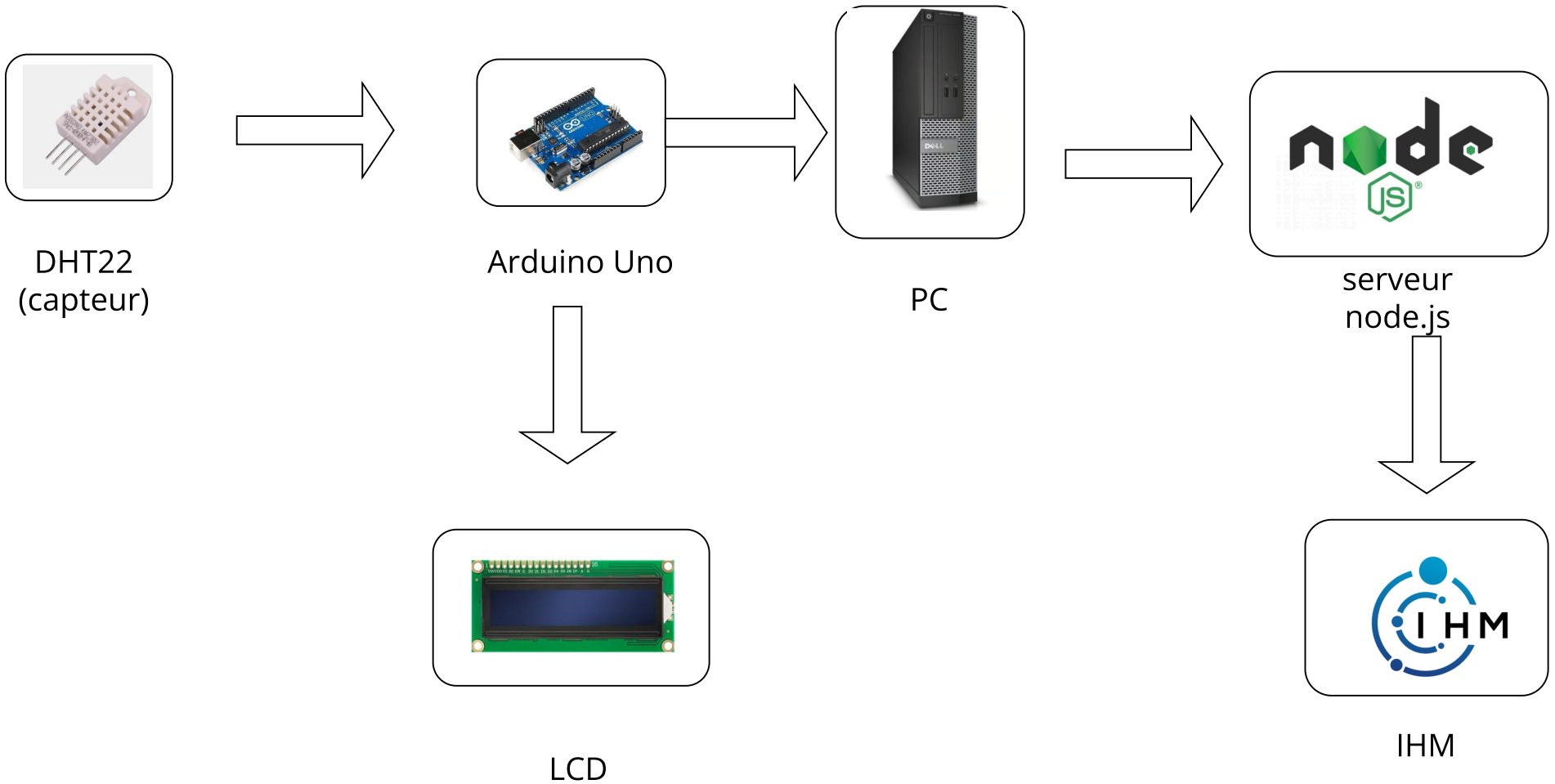


Le diagramme synoptique du système



1. Mesure des Paramètres Environnementaux

- Température : Mesure précise avec une précision de $\pm 0,5^{\circ}\text{C}$.
- Humidité relative : Mesure avec une précision de $\pm 5\%$.

2. Affichage Local

- Visualisation des données en temps réel sur un écran LCD 16x2.
- Rafraîchissement des valeurs toutes les secondes.

3. Transmission des Données

- Envoi des données de température et d'humidité via une communication série (USB) à un PC.

4. Gestion des Données avec un Serveur Node.js

- Réception des données envoyées par l'Arduino.
- Gestion et stockage des données pour consultation en temps réel.

5. Interface Homme-Machine (IHM) Web

- Affichage en temps réel des valeurs mesurées (température et humidité).
- Rafraîchissement automatique toutes les secondes.
- Indication visuelle en cas de dépassement des seuils critiques.

6. Accessibilité Distante

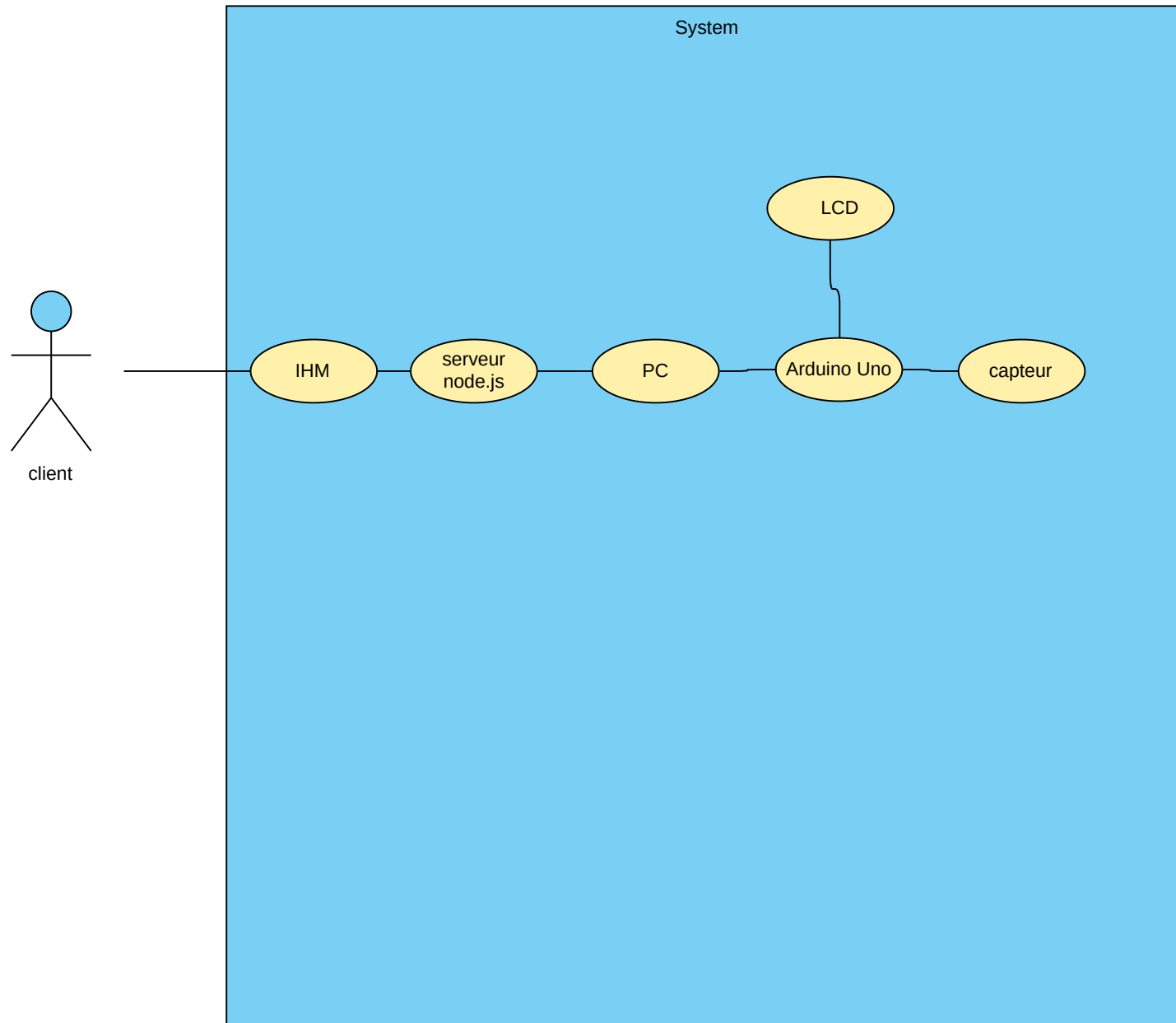
- Consultation des données depuis n'importe quel appareil connecté au réseau local ou distant.
- Interface compatible avec les appareils mobiles et les ordinateurs.



7. Alerte et Surveillance

- Possibilité d'intégrer des notifications ou des alertes (future amélioration).
- Suivi constant pour prévenir les risques de surchauffe ou d'humidité excessive

- Le diagramme des cas d'utilisation



Liste des Exigences du Système

1. Mesure des Données Environnementales

- Utiliser un capteur combiné (par exemple, DHT22) pour mesurer la température et l'humidité.
 - Garantir une précision de $\pm 0,5^{\circ}\text{C}$ pour la température.
 - Garantir une précision de $\pm 5\%$ pour l'humidité relative.
- Effectuer une mesure et un rafraîchissement toutes les secondes.

2. Affichage Local

- Connecter un écran LCD 16x2 (avec interface I2C) pour afficher les données.
- Afficher la température ($^{\circ}\text{C}$) et l'humidité (%) de manière lisible et simultanée.

3. Transmission des Données

- Envoyer les données mesurées en temps réel via une communication série (USB).
 - Utiliser un protocole fiable pour éviter la perte d'informations.

4. Gestion des Données avec Serveur Node.js

- Développer un serveur capable de recevoir et de traiter les données transmises par l'Arduino.
- Héberger une application web permettant la visualisation des données en temps réel.

5. Interface Homme-Machine (IHM) Web

- Afficher les valeurs mesurées (température et humidité) de manière intuitive et claire.
Actualiser automatiquement les données toutes les secondes.
- Fournir des indications visuelles ou sonores en cas de dépassement de seuils critiques.

6. Accessibilité et Surveillance à Distance

- Permettre un accès aux données via un navigateur web sur tout appareil connecté au réseau local ou distant.
- Assurer la compatibilité avec les appareils mobiles et les ordinateurs.

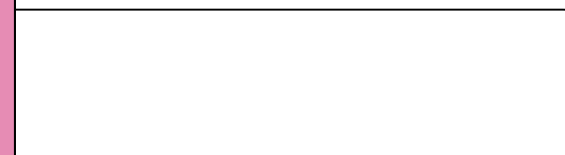
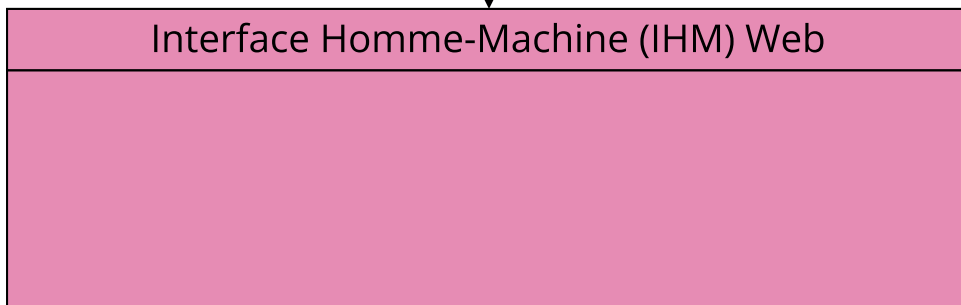
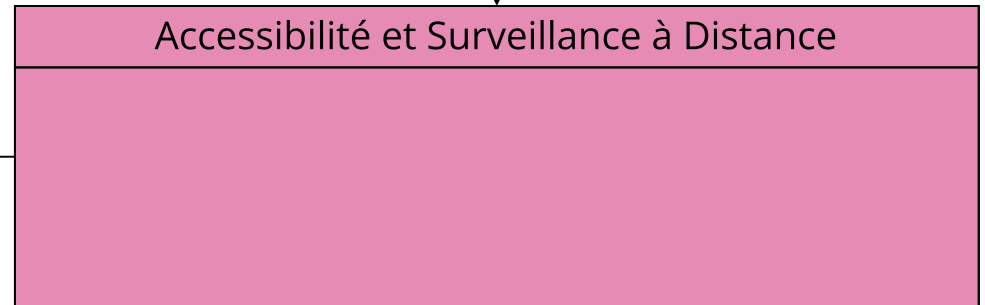
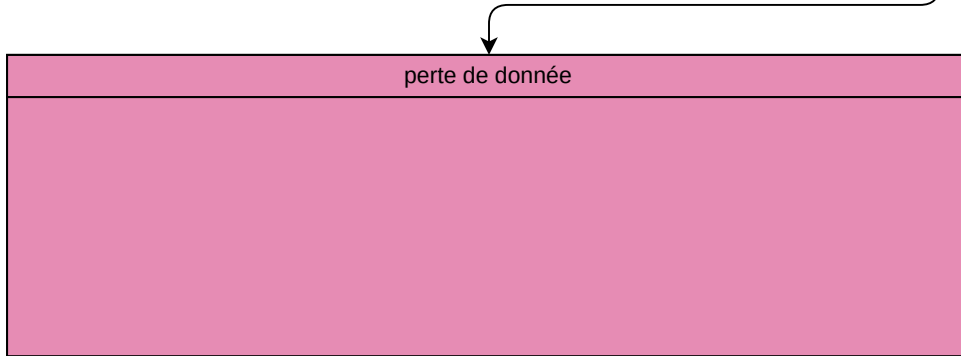
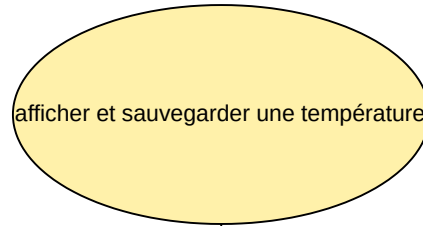
7. Fiabilité et Résilience

- Gérer les erreurs de mesure ou les pertes de données de manière robuste (messages d'erreur clairs en cas de problème).
- Fonctionner en continu pour permettre un suivi constant de l'environnement.

8. Évolutivité et Notifications (optionnel pour l'avenir)

- Prévoir la possibilité d'intégrer des notifications automatiques (email, SMS) en cas de dépassement des seuils critiques.
- Permettre l'ajout de capteurs ou de fonctionnalités supplémentaires si nécessaire

diagramme d'exigence



Gestion des Données avec Serveur Node.js

Transmission des Données

Affichage Local

Mesure des Données Environnementales