

Projet JAVA – Sockets/RMI

Agriconnect

On souhaite mettre en œuvre un réseau de capteurs et d'actionneur pour faciliter le travail d'agriculteurs. Ce système permet, par l'intermédiaire de capteurs, de surveiller deux paramètres essentiels des cultures : l'humidité du sol et la température de l'air. Grâce à l'utilisation d'actionneurs, le système pourra déclencher l'arrosage des cultures si besoin. Le tout pourra être surveillé (et géré) par l'agriculteur sur une application cliente. Vous êtes mandaté pour réaliser une telle application sur la base d'applications lourdes en Java.

Version 1 : Gestion des capteurs

La première version du système se compose d'une centrale de gestion et de capteurs. Chaque capteur prendra la forme d'une application Java et sera identifié par un code unique et ses coordonnées GPS. A intervalle régulier, le capteur remontera à la centrale les informations de température et d'humidité. La centrale de gestion gèrera les capteurs et les informations remontées.

Ajout de capteur :

Lors du démarrage d'un capteur, celui-ci doit se déclarer auprès de la centrale.

Retrait de capteur :

Lors du retrait d'un capteur, celui-ci doit indiquer son retrait auprès de la centrale.

Utilisation courante :

Périodiquement, le capteur remonte à la centrale les informations de température et d'humidité. Celle-ci les enregistre, affiche sur la console les informations du capteur.

Dans cette première version, on se limitera à 5 capteurs, 1 centrale de gestion et l'intervalle de mesure sera fixé à 5 secondes.

Version 2 : Application cliente

Dans cette seconde étape, l'agriculteur doit pouvoir disposer d'une application Java permettant d'interagir avec la centrale de gestion et, par son intermédiaire, les capteurs.

L'application cliente doit permettre :

- De lister les capteurs,
- D'obtenir les dernières informations d'un capteur,
- D'obtenir la moyenne et tendances (à la hausse ou à la baisse) des informations des capteurs sur la dernière heure ou la dernière journée.
- Modification des intervalles de mesure pour 1 capteur ou pour tous les capteurs.
- De déclarer un nouveau capteur ou d'exiger son retrait. En ce sens, cela modifie la gestion de la version 1. Pour être ajouté, un capteur doit être maintenant déclaré (respectivement retiré) par l'application cliente (en utilisant son code unique par exemple). Ainsi, le capteur sera sollicité par la centrale pour débuter l'envoi de ses informations et, respectivement pour le retrait, sera stoppé.

Version 3 : Gestion de l'arrosage

Dans cette troisième version, on met maintenant en place des actionneurs pour l'arrosage. Ceux-ci pourront être déclarés ou retirés sur le modèle des capteurs de la version 2. Toutefois, l'arrosage ne sera commandé que par la centrale de gestion.

Chaque actionneur, comme les capteurs, dispose d'un code unique et de coordonnées GPS.

Vous devez implanter les fonctionnalités supplémentaires :

- Déclenchement manuel de l'arrosage par l'application cliente et ce pour une durée fixée. A la fin de l'arrosage, l'application cliente doit être notifiée. A tout instant, la centrale de gestion doit logger l'état du système d'arrosage (arrosage débuté, arrosage terminé).
- Obtenir l'état de l'arrosage.

Attention, plusieurs capteurs peuvent être impactés par un même actionneur du système d'arrosage. Dans le cadre de ce projet, nous simulerons cela par les hypothèses suivantes :

- Le système d'arrosage « connaîtra » les capteurs et leur communiquera un « +1% » sur le taux d'humidité toutes les 10 secondes.

Organisation du projet et travail attendu

Ce projet est à rendre de manière incrémentale et par groupe de 4. Chaque livrable sera évalué et doit comporter deux parties :

Partie I : Conception de l'application

Concevez et rédigez les interfaces de l'application permettant de répondre aux spécifications précédemment énoncées. Constituez un dossier qui comportera :

- Les diagrammes UML adéquats exprimant le résultat de l'analyse de cette application.
- Les différentes catégories d'entités logicielles en identifiant leur rôle et leur positionnement envisagé sur l'environnement technologique cible.
- Les interactions pouvant survenir entre ces types d'entités.

Partie II : Développement d'une maquette de l'application

Une maquette de la solution envisagée doit être développée. Celle-ci devra donner une image fidèle de la solution réelle ensuite déployée. On s'attachera en priorité (au-delà des aspects IHM, traitement, persistance...) à rendre opérationnelles les interactions entre entités logicielles.

Évaluation

A chaque échéance, chaque **groupe de 4** devra présenter son projet et faire une démonstration. Ce même jour, vous devez rendre le rapport lié à la version livrée. Votre projet sera évalué en fonction des choix techniques que vous effectuerez, de la qualité technique du code produit, de la qualité de la documentation, de son respect du sujet, du respect des échéances, de l'ergonomie de votre application et de son allure générale.

De plus, vous devrez déposer votre rapport et vos sources sur moodle.

Tout retard dans la fourniture des livrables sera sanctionné dans l'évaluation.

Vos livrables pourront être soumis à évaluation d'outils détectant différentes formes de plagiat.

Rappel sur le plagiat :

Il existe de nombreuses bibliothèques et applications qui réalisent tout ou partie du travail demandé. Vous pouvez les regarder et vous en inspirer **à la condition expresse de citer vos sources**. Mais, pour la réalisation de ce projet, il vous est **interdit** :

- de **reprendre du code** de ces bibliothèques ;
- de **s'inspirer**, de **plagier le code** ou la **conception** réalisée par les **autres étudiants**.

Consignes diverses :

- Les fonctionnalités supplémentaires que vous mettrez en œuvre seront aussi comptabilisées **à la condition que tous les éléments du projet aient été correctement traités au préalable ET sont soumises à l'accord préalable de l'équipe enseignante.**