

EC Développement Web

Frédéric Flouvat
Université de la Nouvelle-Calédonie

Plate-forme de gestion et de visualisation de données IoT

- Etude préliminaire (première partie du rapport) avant le 17/09
- Démonstration de 10 minutes, avec 5 minutes de questions, le 8/10
- Rapport final + Code source du projet (fichiers PHP, HTML, CSS, Javascript, images, base de données ...) à rendre le 15/10

1 Le projet

L'objectif du projet est de concevoir une plate-forme de gestion de données IoT (internet des objets). Ce projet consiste à développer un "front-end" Web à des stations de mesure Arduino (datalogger). Une station de mesure fait l'acquisition de données. Ces données sont ensuite envoyées vers une base de données centralisée (dont la structure est à définir). La plate-forme à concevoir a pour but de gérer ces données et d'en permettre leur exploitation pour faire du suivi des mesures collectées.

Ce projet est à réaliser en groupe de quatre étudiants. Il faudra donc collaborer, se répartir le travail, et partager le même code source.

Les sous-sections suivantes décrivent plus en détail des fonctionnalités de la plate-forme. Cette liste de fonctionnalités n'est pas exhaustive. D'autres peuvent être ajoutées si vous le souhaitez.

1.1 Un premier prototype permettant de visualiser les données IoT d'une station de mesure

Dans un premier temps, il faudra développer un prototype basique permettant de visualiser sous forme tabulaire les données collectées par une station de mesure enregistrée sur la plateforme. L'accès aux données sera par défaut limité au propriétaire de la station de mesure. S'il le souhaite, le propriétaire des données pourra rendre publique tout ou une partie de ses données.

1.2 Amélioration de la plate-forme

Le prototype de base permet d'avoir rapidement une solution exploitable mais il reste très pauvre en terme de fonctionnalités. Son amélioration passe par l'ajout d'autres fonctionnalités telles que celles citées ci-dessous.

1.2.1 Gérer plusieurs stations de mesure

La plate-forme pourrait intégrer un espace privé permettant aux utilisateurs de gérer leurs informations et leurs stations de mesure (p.ex. modifier, supprimer). Cet espace permettrait par exemple de lister les stations de mesure enregistrées, de visualiser les données collectées par chacune, et de modifier les informations sur celles-ci (description des stations, liste des capteurs utilisés, localisation, précision des mesures collectées, etc).

1.2.2 Mettre en place un espace administrateur

La plate-forme pourrait intégrer une partie administration. Cette interface permettrait par exemple aux administrateurs de modifier ou de supprimer les informations d'un utilisateur et/ou d'une station.

1.2.3 Afficher des graphiques

Dans le cadre du suivi des mesures collectées, l'utilisateur pourrait générer et afficher des graphiques. Ces graphiques pourraient être dérivés d'un calcul réalisé sur les données brutes (p.ex. calcul de la moyenne, écart-type, etc).

1.2.4 Gérer des projets

Souvent une station de mesure n'est pas issue d'un développement isolé. Elle est associée à un projet plus global (p.ex. suivi de la qualité de l'air dans une ville) comprenant un ensemble de stations de mesure similaires localisés à différents endroits. Dans ce cadre, la plate-forme pourrait permettre de créer des projets et d'y associer un ensemble de stations (tout comme pour un utilisateur). Contrairement à un compte utilisateur, les informations d'un projet (p.ex. nom du projet, personnes impliquées, institutions partenaires etc) seraient publiques. Les données collectées seraient par défaut réservées aux membres de celui-ci (avec la possibilité de rendre tout ou une partie publique). Tout utilisateur pourrait s'inscrire (collaborer) à des projets. Cette inscription donnerait l'accès à l'intégralité des données collectées. Elle donnerait aussi accès au plan de montage des stations de mesure et aux codes sous-jacents (cf site <https://www.hackster.io>), permettant ainsi à l'utilisateur de développer sa station et de générer de nouvelles données pour le projet.

1.2.5 Afficher la localisation des stations sur une carte

La plate-forme pourrait intégrer une visualisation cartographique des stations de mesure (p.ex. via Google Map). Cette carte permettrait de visualiser la localisation des différentes stations et de visualiser les données associées à celles-ci.

1.2.6 Filtrer les données

La plate-forme pourrait intégrer un système de filtres afin de pouvoir cibler les données affichées. Le filtrage des données pourrait se faire en fonction du type de mesures (p.ex. météo, qualité de l'air, etc), de la date, de la localisation géographique ou des valeurs. Il pourrait être intéressant de pouvoir enregistrer ces filtres comme une nouvelle source de données.

1.2.7 Partager des données privées

Un utilisateur pourrait demander au propriétaire d'une ou de plusieurs stations l'accès à des données non publiques. Le propriétaire recevrait alors une notification de cette demande (p.ex. via son espace personnel ou un mail). Il pourrait répondre au message et valider l'inscription de l'utilisateur à cette source de données.

1.2.8 Mettre en place des alertes

Un utilisateur pourrait mettre en place des alertes en cas de dépassement de certaines valeurs. Ces alertes pourraient se présenter sous la forme de messages dans l'interface privée des utilisateurs et/ou sous la forme d'envoi de mails.

1.2.9 Exporter des données

La plate-forme pourrait permettre l'exportation de données au format CSV.

2 Evaluation du projet

Ce projet est associé à deux notes de CC. La première note évaluera la documentation réalisée et la deuxième note évaluera l'application développée à proprement parlé.

Une partie de l'évaluation sera faite par les membres du groupe ("peer evaluation" ou évaluation par les pairs). Plus précisément, les membres de chaque groupe se répartiront 8 points pour chaque "livrable" (la documentation et l'application). Par exemple, si tous les membres d'une équipe de 4 ont travaillé équitablement, ils pourront se mettre 2 points chacun. Si un des membres a plus contribué que les autres, l'équipe peut décider de lui affecter 3 points, et se répartir les 5 points restants.

2.1 La documentation

La documentation (document au format pdf) décrira les fonctionnalités développées, ainsi que la façon dont elles ont été développées. Ce rapport devra donc aussi décrire les différentes étapes du processus de développement. Il sera décomposé en deux parties.

La première partie présentera l'étude préliminaire, c'est-à-dire la définition du projet (besoins, profils des utilisateurs, scénarios d'utilisation), le design de l'interface utilisateur (diagrammes d'expérience utilisateur, squelettes graphiques, maquettes graphiques), le design de l'architecture interne (schéma de bases de données, architecture du code, politique de sécurité, technologies utilisées), la liste des fonctionnalités à implémenter (et leur priorité), le calendrier et l'organisation des "sprints".

La deuxième partie décrira la réalisation du projet, c'est-à-dire le détail de chaque "sprint" (les fonctionnalités prévues et celles implémentées, les interfaces graphiques développées, l'architecture du code finalement choisi, les tests implémentés, et les différentes versions obtenues). Une description d'outils et de bonnes pratiques pour faire ce type de documentations peut être trouvée sur le site suivant :

<https://www.altexsoft.com/blog/business/technical-documentation-in-software-development-types-best-practices-and-tools>.

Vous pourrez aussi trouver quelques exemples de rapports des années précédentes sur le Moodle du cours.

2.2 L'application

Dans un premier temps, l'application sera évaluée lors d'une démonstration (10 minutes de présentation et 5 minutes de questions). Cette démonstration visera à présenter les différentes fonctionnalités développées, un peu comme vous les présenteriez à un "client" susceptible d'acheter l'outil développé. Il est fortement conseillé de préparer un scénario de démonstration à l'avance et de le répéter plusieurs fois avant le jour de la démonstration.

Dans le second temps, le code source du projet sera aussi analysé par le responsable de cours. Ainsi, le projet sera non seulement évalué en terme de fonctionnalités développées, mais aussi de qualité du code produit (p.ex. indentation, commentaires, architecture MVC, sécurité). L'utilisation de bibliothèques ou d'API (p.ex. pour afficher des cartes ou des graphiques) sera aussi valorisée.