



Analysis Report

Equipe Projet

Axel BERNARD
Moussa BERTHE
Yao KONAN

Sommaire:

I. Rappel du Besoin et Critères de Succès	2
1. Besoin	2
2. Critères de succès	2
II. Model du domaine métier	2
1. Diagramme des cas d'utilisation	2
2. Diagramme de Séquence Système	3
3. Diagramme de Classe	5
III. Description de l'Écosystème et Contraintes	6
1. Description de l'Écosystème	6
2. Contraintes	6
3. Principe de Solution	6

I. Rappel du Besoin et Critères de Succès

1. Besoin

Avec l'accélération du développement logiciel et l'augmentation de l'utilisation des pipelines CI/CD, il est nécessaire de disposer d'une solution pour mesurer la consommation d'énergie d'un build. De plus, un outil capable de prédire la consommation d'énergie d'un projet et fournir des conseils aux utilisateurs est également requis. C'est face à ce défis que nous avons décidé de mettre en place un plugin Jenkins capable de suivre la consommation d'énergie d'un projet et d'y associer à un Linter VSCode pour prédire cette consommation afin d'enseigner les bonnes pratiques de développement aux utilisateurs pour la minimiser.

2. Critères de succès

Pour que le projet soit considéré comme une réussite il faut:

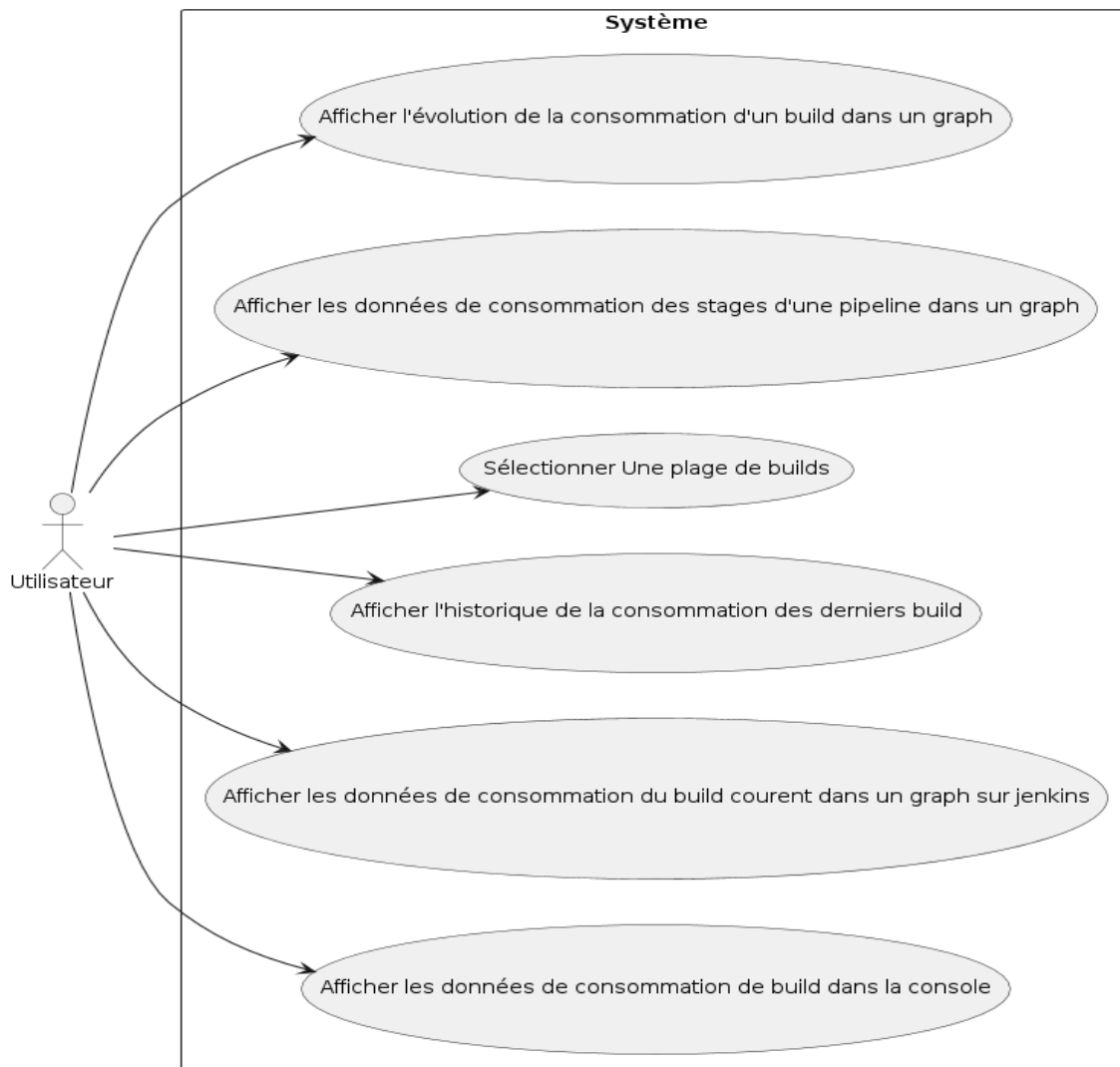
- Des mesures précises de la consommation d'énergie
- La visualisation de l'historique de la consommation d'énergie dans SonarQube ou autres
- Une assistance efficace aux développeurs pour l'application de meilleures pratiques.
- Une prédiction approximée de la consommation d'énergie d'un projet

II. Model du domaine métier

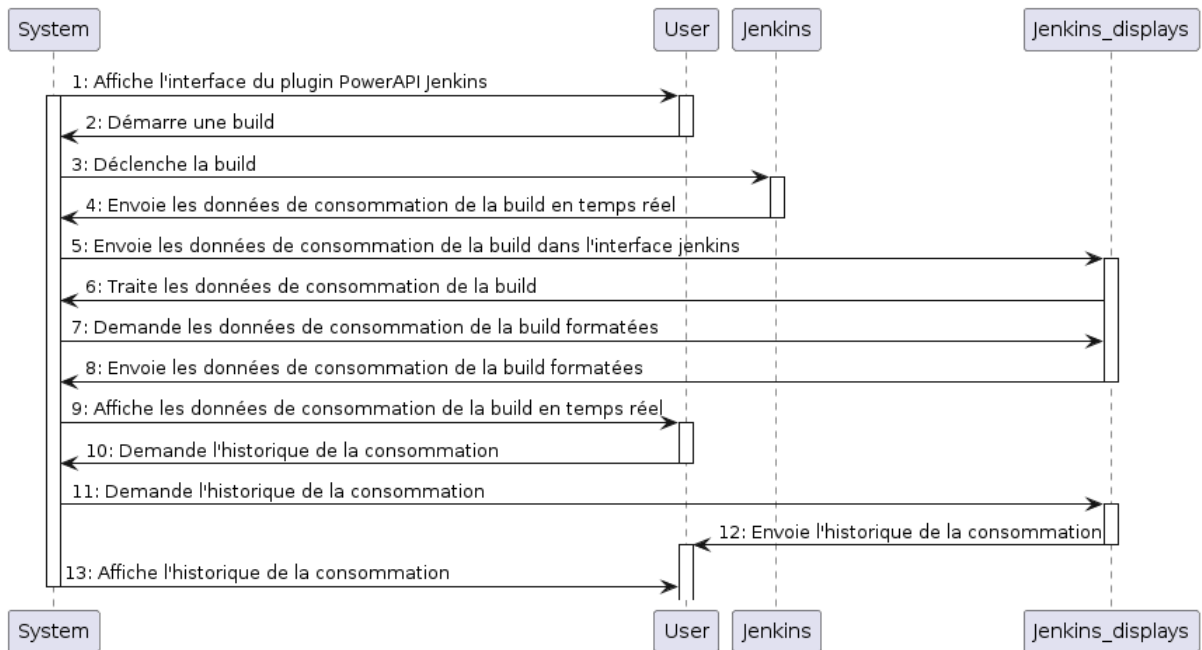
1. Diagramme des cas d'utilisation

- **Afficher l'évolution de la consommation d'un bâtiment dans un graphique:** Cette action permet à l'utilisateur de visualiser l'évolution de la consommation d'énergie d'un build sur une période donnée.
- **Afficher les données de consommation des stages d'une pipeline dans un graphique:** Cette action permet à l'utilisateur de visualiser les données de consommation d'énergie des différentes étapes d'une pipeline jenkins.
- **Afficher l'historique de la consommation des derniers builds:** Cette action permet à l'utilisateur de visualiser l'historique de la consommation d'énergie des derniers builds.
- **Afficher les données de consommation du build courant dans un graph sur jenkins:** Cette action permet à l'utilisateur de visualiser les données de consommation d'énergie du build en cours d'exécution sur Jenkins.

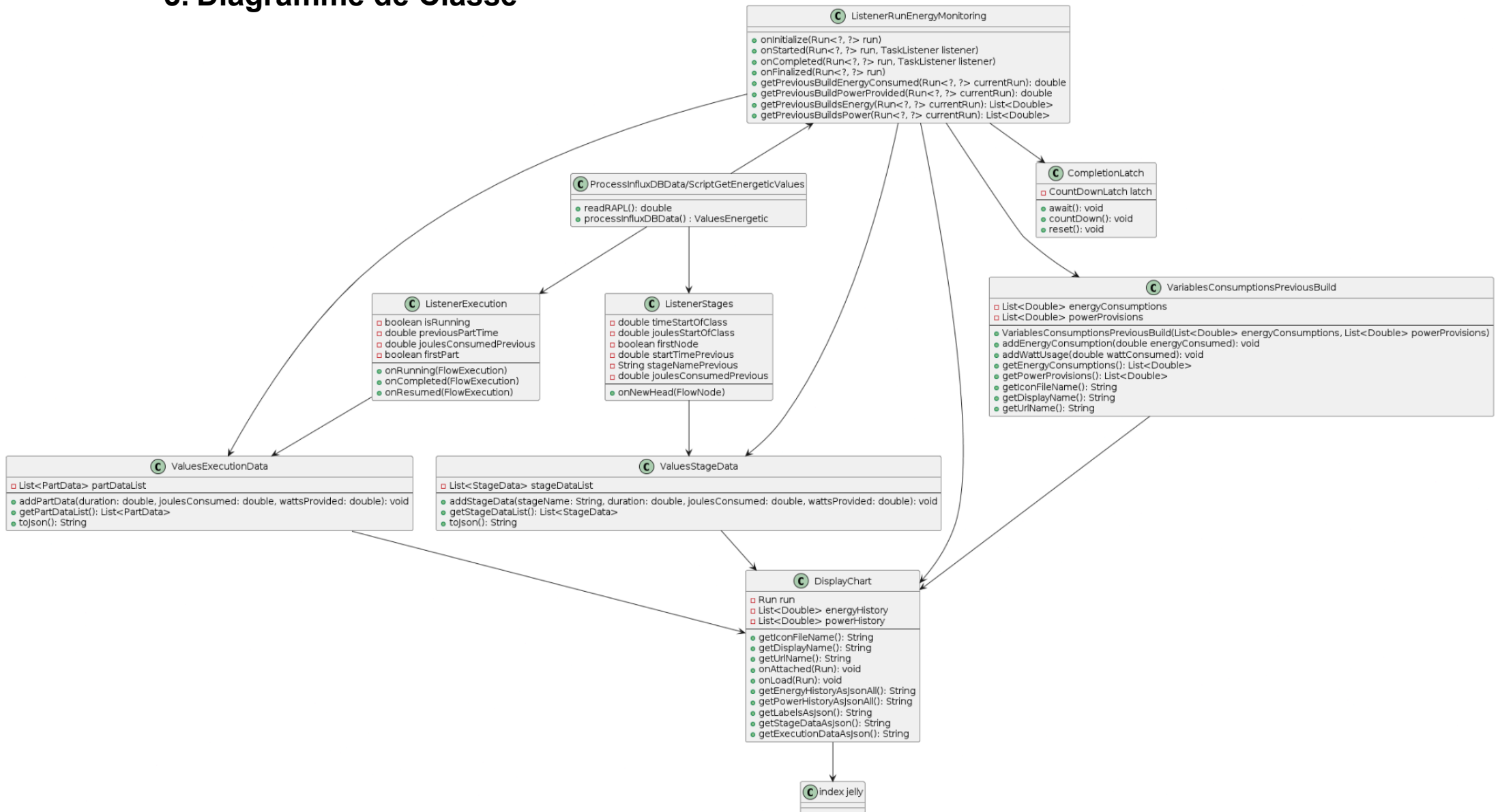
- **Afficher les données de consommation de build dans la console:** Cette action permet à l'utilisateur d'afficher les données de consommation d'énergie d'un build dans la console



2. Diagramme de Séquence Système



3. Diagramme de Classe



III. Description de l'Écosystème et Contraintes

1. Description de l'Écosystème

Notre système s'intègre dans un écosystème où le développeur utilise Jenkins et Visual Studio Code comme environnement de travail. Pour mesurer la consommation énergétique d'un Jenkins, notre système doit être utilisé avec cet outil. Par ailleurs, Visual Studio Code est également essentiel à notre solution, car nous visons à développer une extension pour cet éditeur.

2. Contraintes

- Nos outils (plugin, extension VSCode) doivent pouvoir être facilement téléchargeables et configurables par l'utilisateur,
- L'utilisateur doit pouvoir facilement visualiser ses données de manière compréhensive pour lui permettre d'avoir un retour clair sur sa consommation, ou sur les prédictions de sa consommation,
- Les mesures de la consommation d'énergie et les prédictions ne doivent pas impacter significativement les performances des pipelines CI/CD,
- Le système doit être compatible avec les outils de CI/CD existants (comme Jenkins) et doit s'intégrer de manière transparente dans le flux de travail des développeurs,
- Nous devons fournir une documentation complète et claire pour permettre aux développeurs de comprendre le fonctionnement de notre outil et de l'utiliser facilement.

3. Principe de Solution

Notre solution se divise en deux parties qui sont:

- Un plugin Jenkins : Les utilisateurs peuvent l'intégrer à un pipeline, activant ainsi le plugin à chaque build de cette pipeline. Lors du build, la consommation d'énergie est affichée dans la console et disponible sous forme de graphe dans un onglet de Jenkins. De plus, les utilisateurs peuvent accéder à un graphique de l'historique de leur consommation via un autre outil de visualisation.
- Un linter pour VSCode : Téléchargeable depuis Visual Studio Code, ce linter prédit en temps réel la consommation énergétique d'un projet. Il fournit également des conseils et des bonnes pratiques à appliquer dans le code pour réduire la consommation énergétique.