Admineasy

Cahier des charges - Novembre 2017

Antoine Dujardin

Chloé Trugeon

Kylian Bohl

Mélissa Da Costa

Table des matieres

Description du projet

Nom du groupe	MACK
Membres	Mélissa DA COSTA Chloé TRUGEON Antoine DUJARDIN Kylian BOHL
Responsable	Kylian BOHL
Nom du projet	Admineasy
Résumé	Le projet est une application dédiée aux administrateurs réseau. Elle leur permettra de consulter l'état de leur machines. Nous développerons une application client (Windows + Linux) pour récupérer les données, une application serveur, et un site internet affichant les données, voire permettant d'effectuer des opérations sur les machines.
Tuteur	Régis Brouard

Etude de l'existant

Présentation générale de l'établissement

Admineasy a pour but d'aider les administrateur réseau a lister tous les ordinateurs présent sur leur réseau, et connaître leur état. Il serait utilisé dans tous types d'entreprises.

Étude de l'environnement

Bien que ce ne soit pas le métier principal de l'entreprise, administrer un réseau est nécessaire dans toutes les entreprises. Une application comme AdminEasy participe donc à un large environnement de métiers, notamment dans les secteurs secondaire et tertiaires. Elle se place principalement dans un marché B TO B (business to business), mais peut azussi être dans un marché B TO C (business to client).

État des lieux

Avec l'environnement concurrentiel de Porter, on peut analyser plusieurs aspects en lien direct avec la concurrence comme:

Pouvoir de négociation des clients

En effet, les administrateurs réseaux étant nombreux, ils exercent une certaine force qui oriente les créateurs d'applications sur ce qu'ils désirent.

Pouvoir de négociation des fournisseurs

Dans notre cas, les "fournisseurs" sont les développeurs qui créent l'application qui est ensuite ajoutée sur le marché. Ici, pour administrer un réseau, plusieurs fonctionnalitées peuvent être misent en place et les administrateurs réseaux n'ont pas tous les mêmes besoin. Certains souhaitent avoir accès à des fonctionnalités spécifiques ce qui n'est pas toujours disponible, ils sont alors contraints d'utiliser une application qui ne leurs convient pas entièrement. Donc cela donne plus de pouvoir aux développeurs qui auront toujours des clients car les administrateurs réseau sont obligés de se servir de certains outils pour exercer leur travail.

Menace des produits ou services de substitution

Plusieurs systèmes de surveillance d'ordinateurs pour entreprise existe déjà, ce qui laisse aux administrateur réseau un certain choix. Ils ont donc la possibilitées de changer d'application.

Parmi ces choix, on retrouve les suites Nagios et Shinken qui permettent d'effectuer le monitoring des systèmes, des protocoles, des applications ou des bases de données notamment.

Il existe encore un très grand nombre de logiciels similaires comme on peut le voir sur ces recherches google :

https://www.google.fr/search?q=enterprise+monitoring+tools

https://www.google.fr/search?q=enterprise+ticketing+systems

Menace d’entrants potentiels sur le marché

Sachant que les besoins des administrateurs réseaux peuvent évoluer rapidement aux fils des avancées technologique et que le monde de l’informatique étant très mouvant, il peut y avoir à tout moment une nouvelle application qui répond aux besoins des administrateurs réseaux.

Degré de concurrence

Comme dit précédemment, les besoins des administrateurs réseau peuvent varier d’un administrateur à l’autre. C’est pour cela qu’il n’y a pas une très grande rivalité (pour le moment) car il n’existe pas deux applications très similaire.

Analyse des besoins

Description des besoins

Le besoin principal est la surveillance des ordinateurs. L'administrateur devra pouvoir accéder aux caractéristiques de tous les ordinateurs dans son réseau, ainsi que leur utilisation. L'administrateur pourra aussi définir des alertes, par exemple si un ordinateur chauffe trop.

Si nous avons assez de temps, nous ajouterons un système de rapport d'erreur. Les utilisateurs pourront créer des rapports d'erreurs, qui seront envoyés à l'administrateur. Cette fonctionnalité n'est pas essentielle à notre application.

Définition de l'objectif du projet

Surveillance

Il faudra lister tous les ordinateurs connectés à un réseau local, et récupérer des informations sur ces ordinateurs. Il y a deux types d'informations: les informations basiques, qui changent rarement, et les informations sur l'utilisation, qui peuvent changer chaque seconde.

Les informations basiques incluent le nom de l'ordinateur, son adresse IP, le nom et la version du système d'exploitation, le nombre de coeurs, etc.

Les informations d'utilisation incluent la température et l'utilisation du processeur, l'espace disponible sur le disque dur, etc.

Interface Web

L'interface web vas donner la liste de tous les ordinateurs connectés au réseau.

Elle vas donner plus d'informations sur les ordinateurs qui ont le client installe. Cela permettra de connaître facilement la configuration des ordinateurs, et leur utilisation. Ces informations pourrait aider à détecter des virus, ou des mineurs de crypto monnaies installés sans autorisation.

L'administrateur devra se connecter pour accéder à l'interface web.

Alertes

L'administrateur pourra définir des alertes en fonctions de certains critères, comme la température CPU d'un ordinateur par exemple. Ces alertes pourront être envoyées par email ou directement sur un smartphone comme une notification, en utilisant des services comme Pushover. Bonus: exécuter des scripts si une condition est remplie

Rapports d'erreur

Les utilisateurs pourront envoyer des messages à l'administrateur depuis le client. L'administrateur pourra les afficher sur l'interface web. Une fois résolu, un rapport d'erreur pourra être fermé par l'administrateur.

Description de la solution

Caractéristiques fonctionnelles

Client

Le client est programmé en Python. Les informations sur le systèmes sont obtenues à partir de librairies externes. Nous avons préféré nous concentrer sur la logique du programme plutôt que chercher où se trouve les informations sur chaque version de chaque système d'exploitation.

Surveillance

Les informations basiques sur chaque ordinateur seront envoyées à la base de données principale.

Les informations sur l'utilisation de l'ordinateur sont mise à jour sur une autre base de données. La fréquence de rafraîchissement vas varier en fonction de la donnée. Par exemple, la température sera vérifiée plusieurs fois par minute, mais l'utilisation du disque dur sera rafraîchie moins souvent.

Contact

Si nous avons le temps d'ajouter cette fonctionnalité, Les utilisateurs pourront soumettre un rapport d'erreur directement depuis l'application.

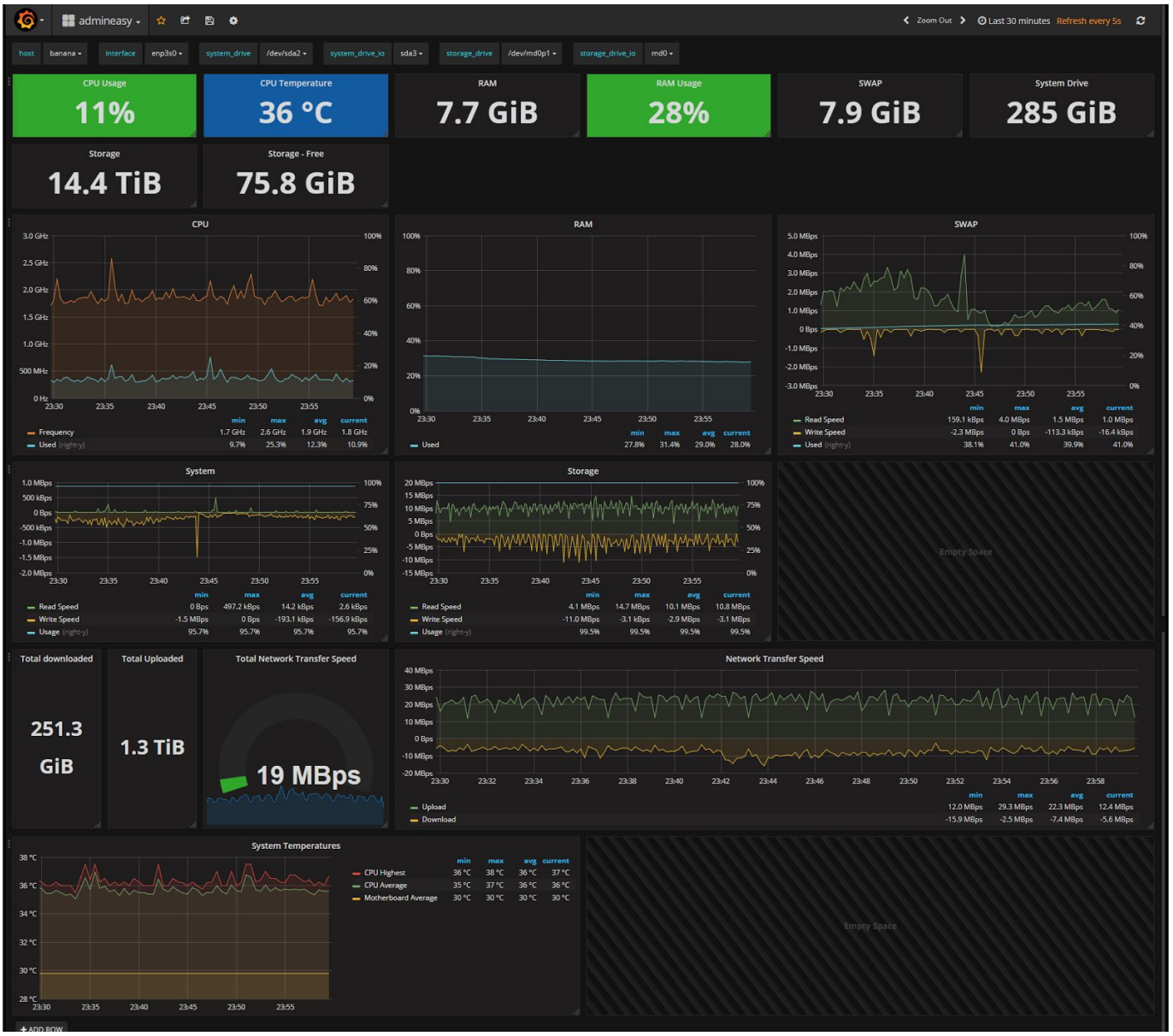
Serveur

Le serveur va lister tous les ordinateurs présents sur le réseau local. Cela inclut les machines sur lesquelles le client n'est pas installé.

Il pourra aussi envoyer les notifications.

Interface Web

Si nous avons assez de temps, nous ajouterons des graphes d'utilisation pour les différents ordinateurs, comme avec Grafana:

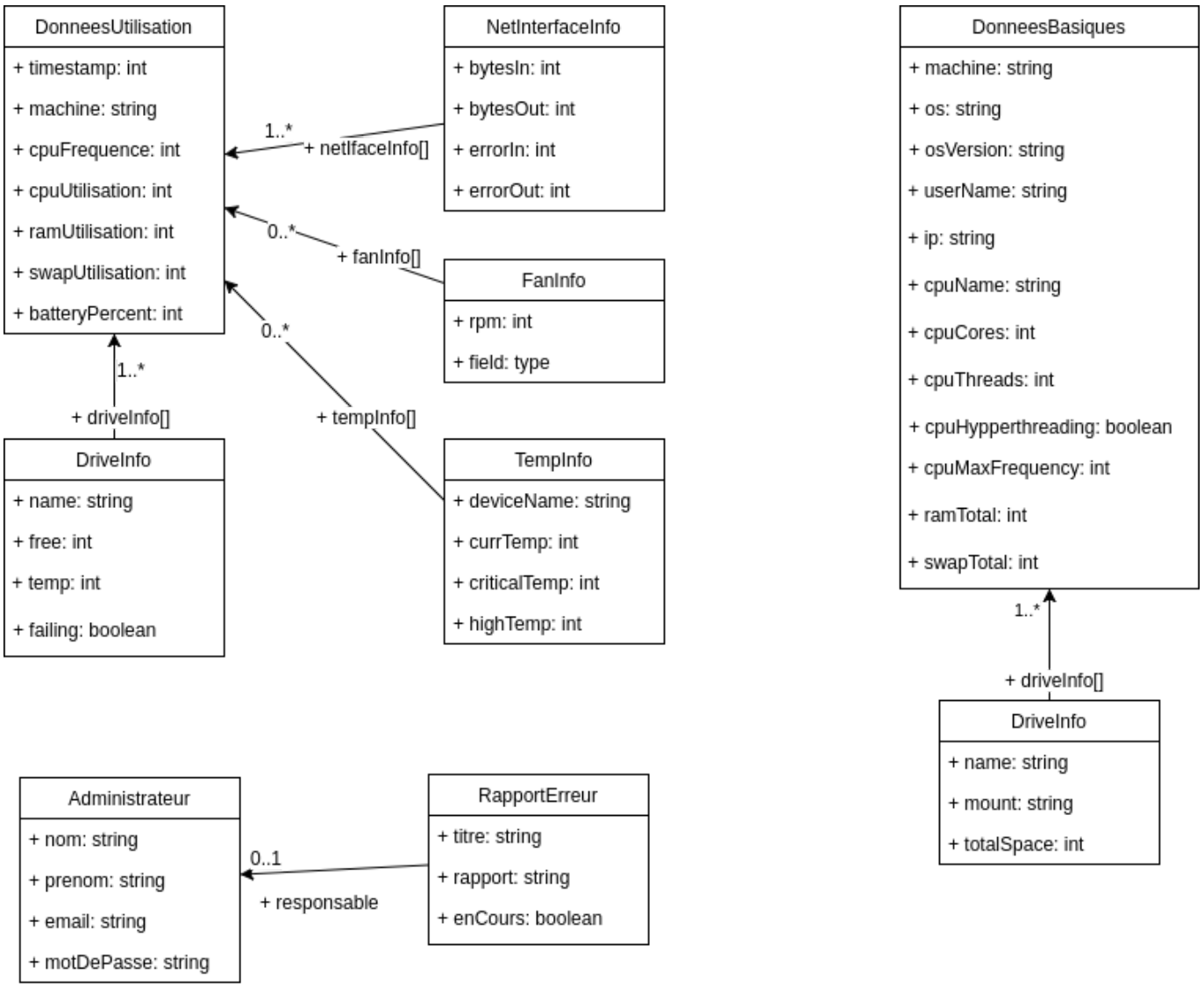


Si nous avons le temps pour ajouter un système de rapport d'erreurs, les rapports d'erreurs seront aussi visible. L'administrateur pourra marquer les rapports sur lesquels il travaille.

Diagrammes UML

Bases de donnees

Diagramme de classes



Client

Diagramme de cas d’utilisation

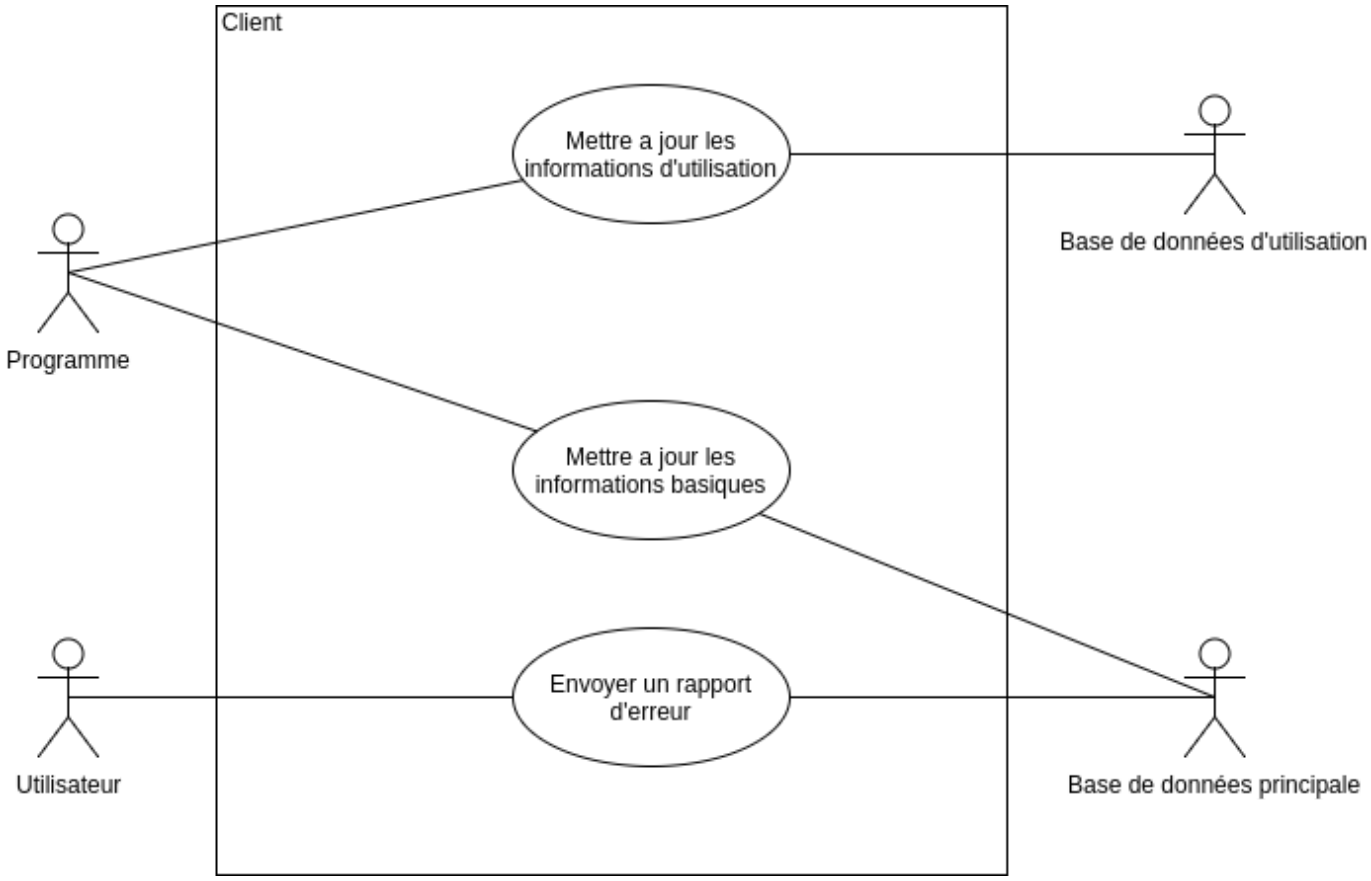


Diagramme de séquence

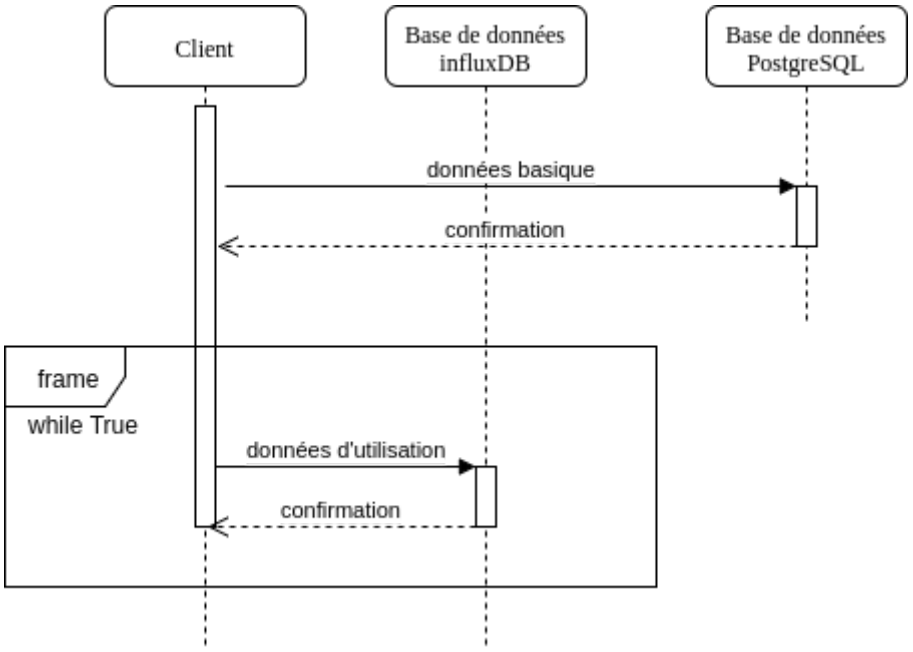
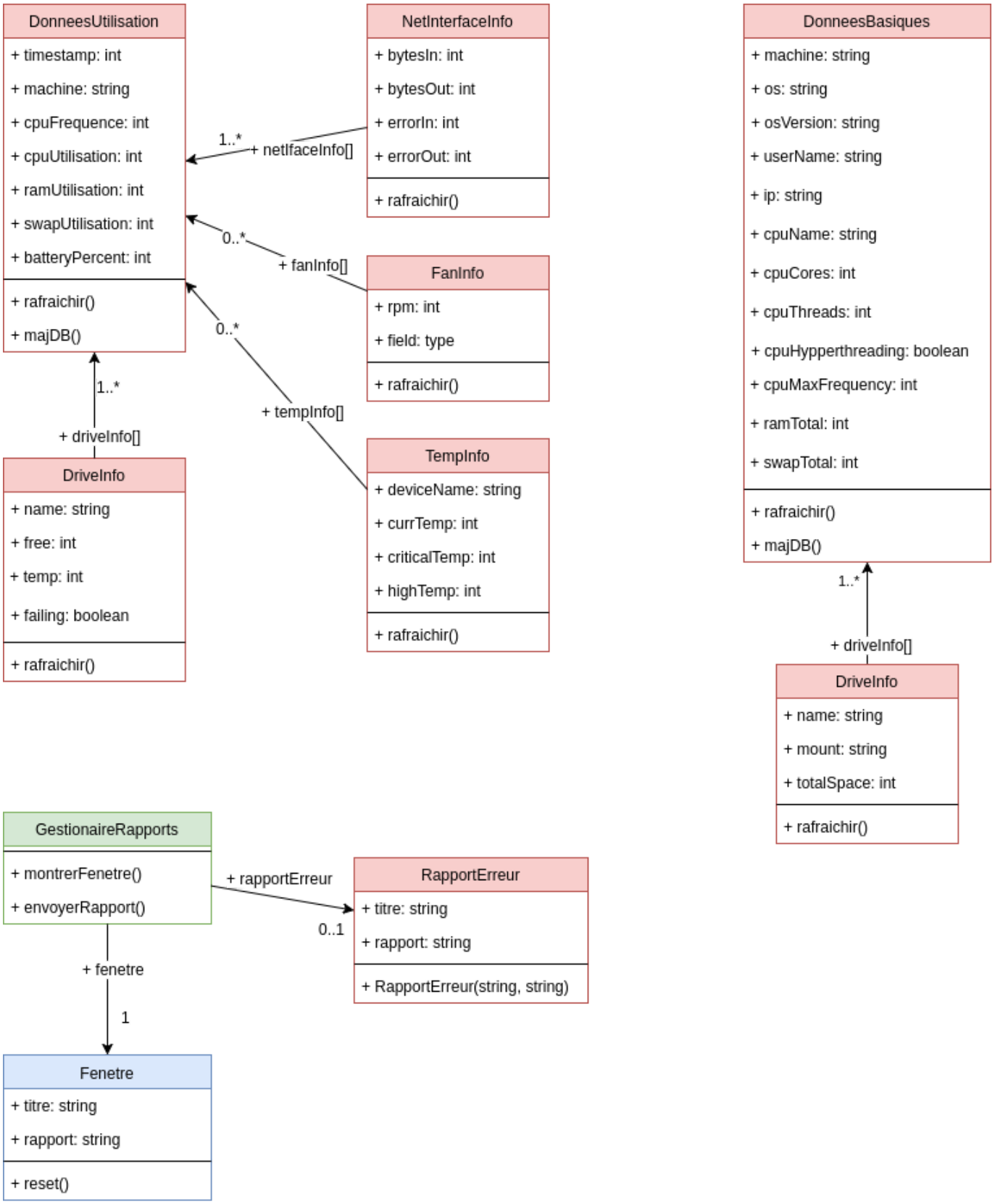


Diagramme de classes



Interface Web

Diagramme de cas d'utilisations

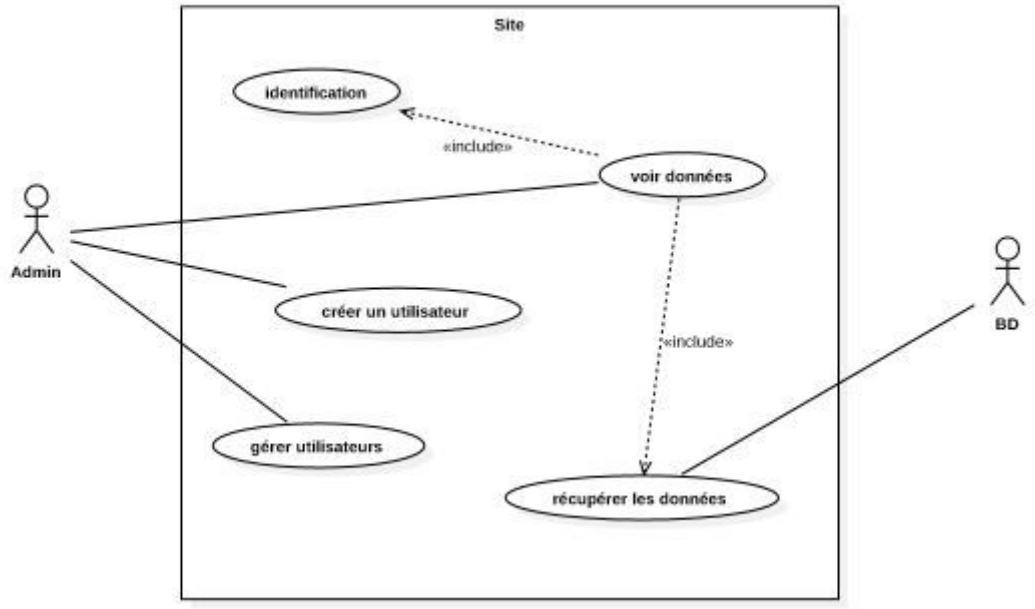
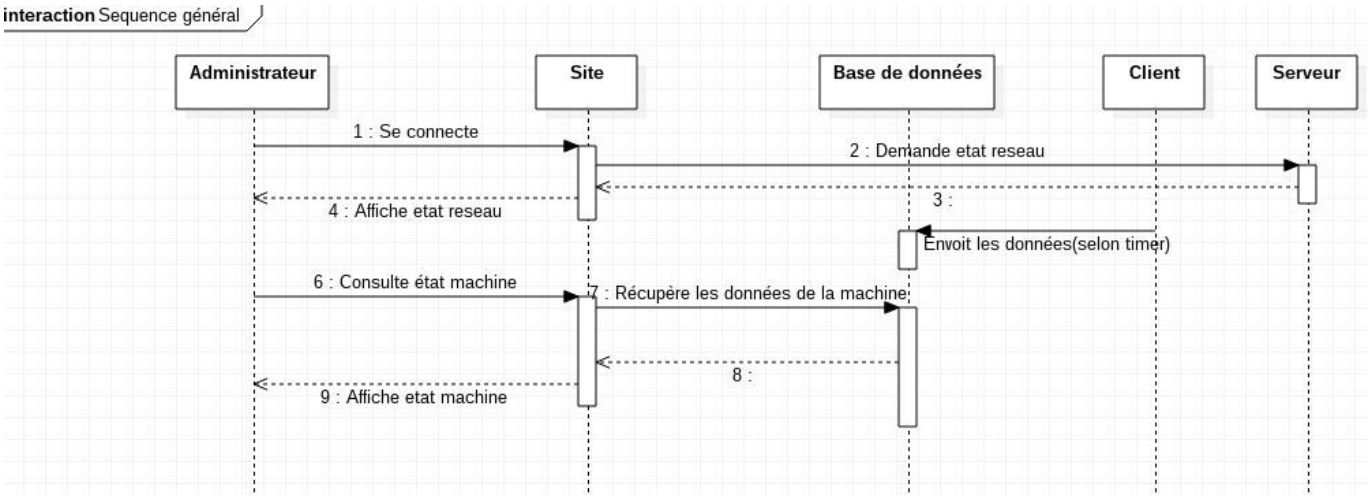


Diagramme de séquence



Outils et Logiciels

Bases de donnees

Nous utiliserons Influxdb, une base de données pour données temporelles, pour stocker les informations sur l'utilisation des ordinateurs.

Nous utiliserons une base de données PostgreSQL pour le reste des données.

Outils

Nous utiliserons des emails et Pushover pour transmettre des alertes. Nous pourrons aussi ajouter des services comme Pushalot ou PushBullet par exemple, leur fonctionnement est très similaire.

Nous pourrons utiliser Grafana pour vérifier facilement l'état de nos bases de données.

Librairies

Client

- Psutil: informations sur le matériel et son utilisation. License BSD.
- Py-cpuinfo: informations détaillées sur le modèle de processeur. License MIT.
- pySMART: analyse du SMART (informations sur l'état des disques dur). License GNU GPL.
- Influxdb: communication avec des bases de données Influxdb. License MIT.
- Psycopg2: communication avec des bases de données PostgreSQL. Licence LGPL.

Définition de la procédure

Découpage en lot

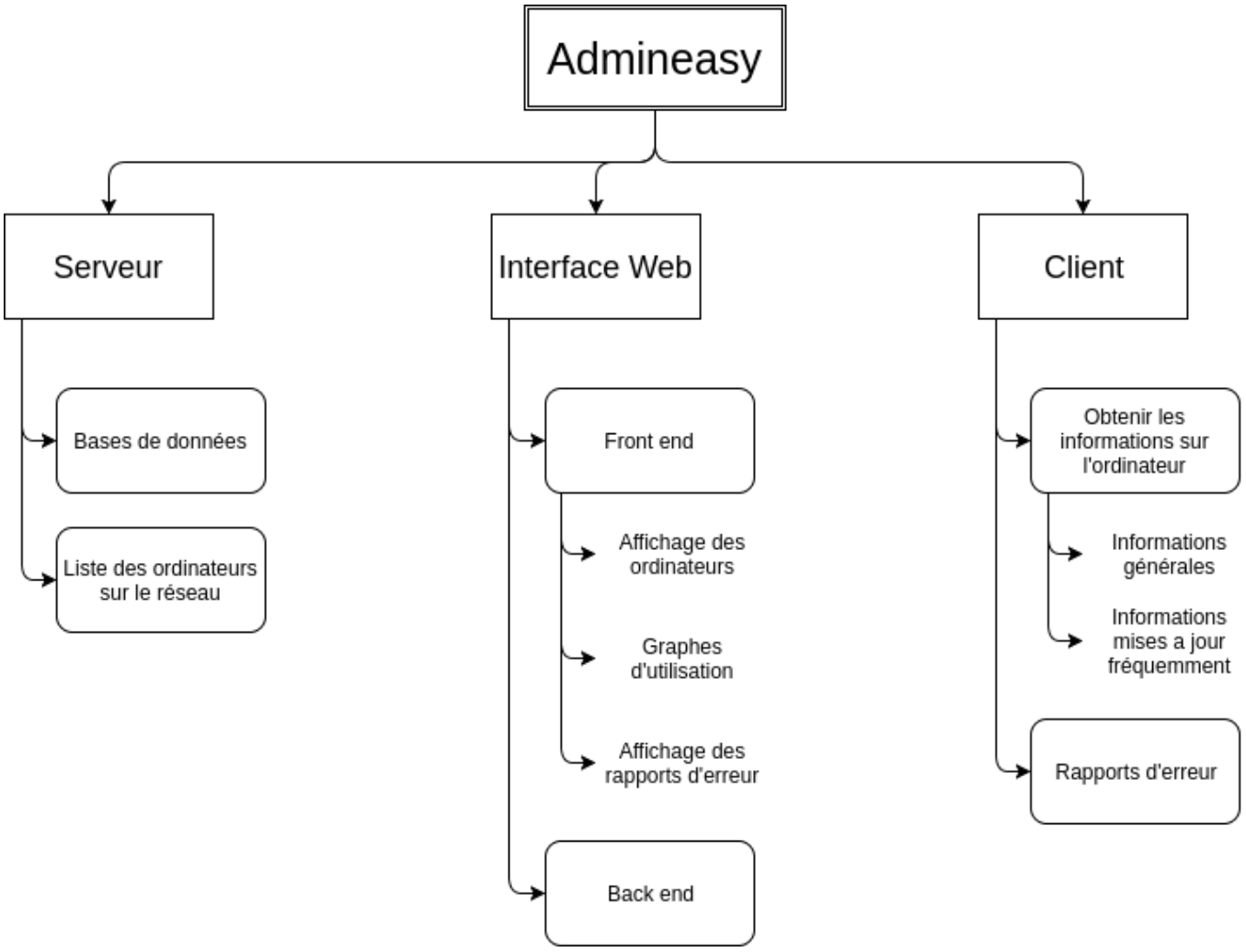


Diagramme de GANTT

Disponible ici (version en ligne)