

Rapport intermédiaire

Table des matières

1. Présentation de l'équipe.....	2
2. Présentation du projet.....	2
3. Uses Cases	2
4. Diagrammes de classe	3
4.1. Module de configuration	5
4.2. Module de supervision.....	5
4.3. Module des logs	5
4.4. Conclusion du diagramme de classe	6
5. Technologies utilisées	6
5.1 Technologies logicielles.....	6
5.2 Technologies web	6
5.3 Stockage des données	7
6. Plan de développement.....	Erreur ! Signet non défini.
7. Gantt & avancement du projet	8
8. Bilan de mi-projet.....	9

1. Présentation de l'équipe

- Développeur :
 - LAKOMY Alexandre
 - PETIT Clément
- Chef de Projet / Client :
 - LE FLOUR Thierry
- Lien GitHub :
 - <https://github.com/AlexandreLakomy/ETRS011>

2. Présentation du projet

L'objectif de ce projet est de concevoir un outil de supervision réseau basé sur le protocole SNMP. Cet outil permettra de surveiller en temps réel un ensemble d'équipements réseau via une interface web offrant des visualisations graphiques et textuelles. Chaque équipement pourra être supervisé selon un niveau de détail défini (CPU, RAM, trafic, stockage, etc.).

Le système sera organisé autour de trois modules principaux :

- **Module de configuration** : permet l'ajout, la suppression et la modification des appareils à superviser.
- **Module de surveillance** : gère le requêtage SNMP et la collecte des données en temps réel.
- **Module de log** : enregistre les événements, messages d'erreur et historiques de supervision.

Deux types d'utilisateurs auront accès à l'application : administrateur et utilisateur standard, chacun disposant de droits spécifiques.

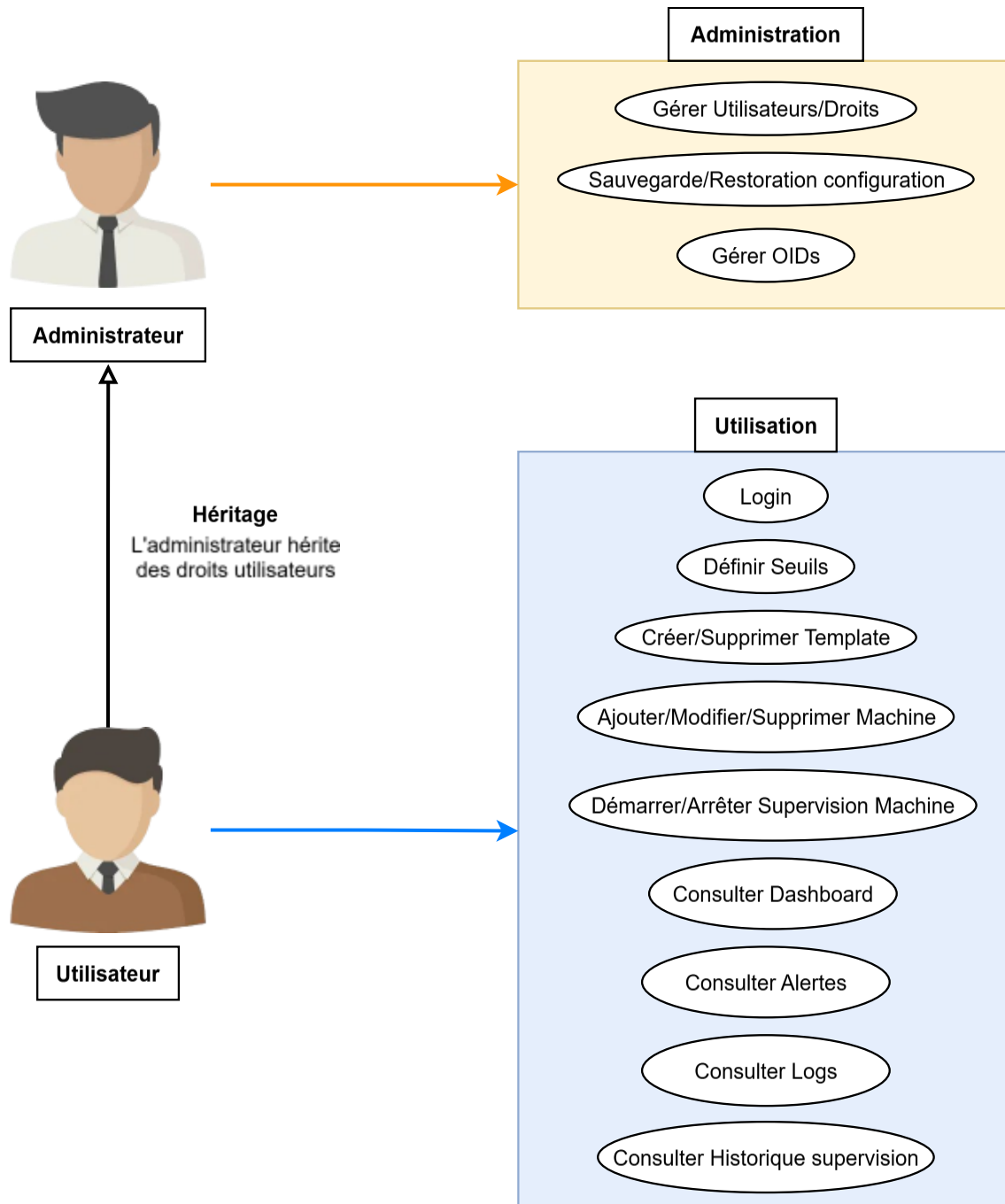
3. Uses Cases

Droits utilisateur :

- Connexion (login)
- Définition d'un seuil
- Créer/Supprimer template
- Ajouter/Modifier/Supprimer Machine
- Démarrer/Arrêter Supervision Machine
- Consulter Dashboard
- Consulter Alertes
- Consulter Logs
- Consulter Historique supervision

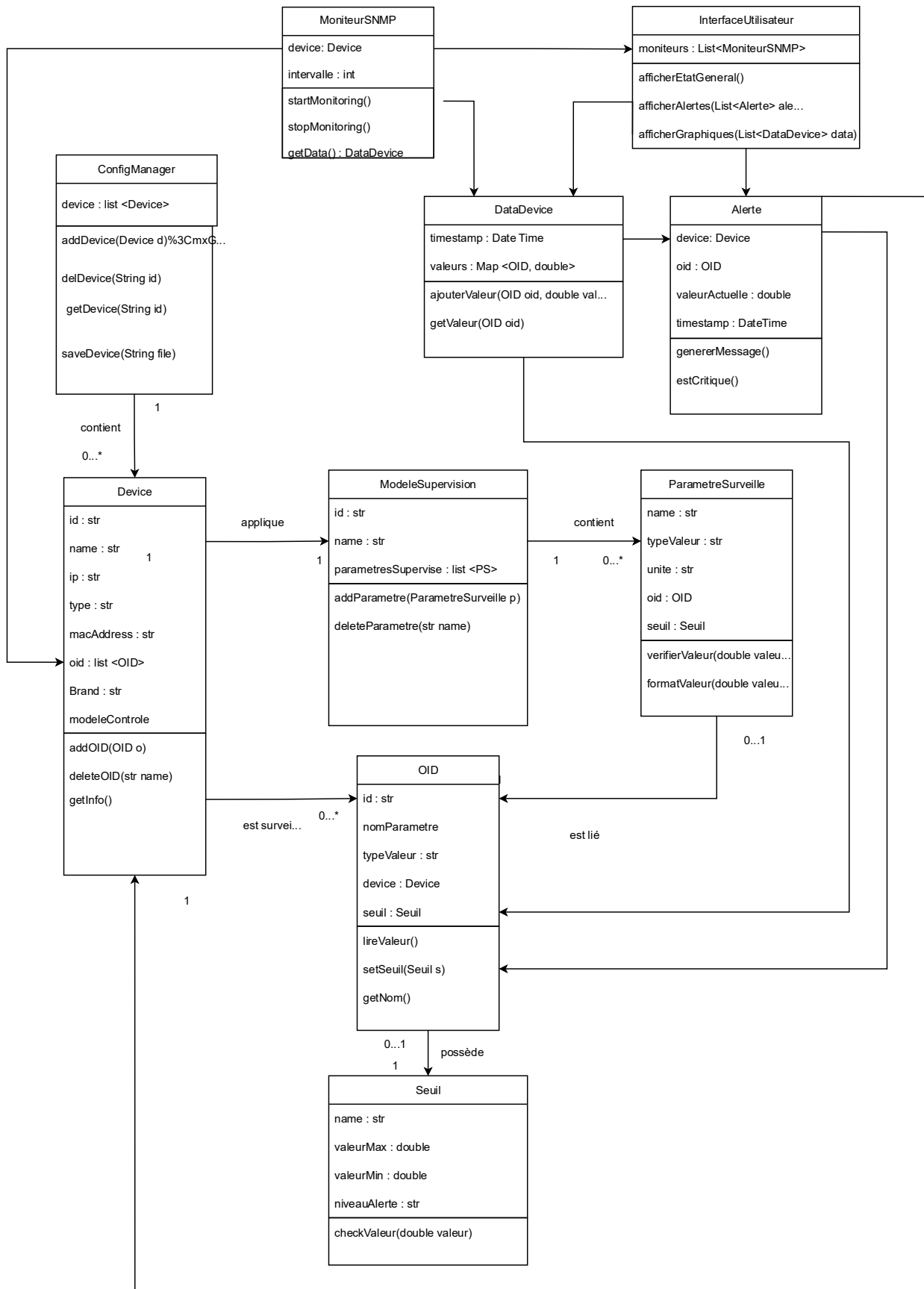
Droits administrateur :

- *Hérite de tous les droits de l'utilisateur*
- Gérer Utilisateurs/Droits
- Sauvegarde/Restauration configuration
- Gérer OIDs



4. Diagrammes de classe

Le diagramme de classes ci-dessous présente la structure générale du système de supervision développé. Il met en évidence les principales entités logicielles, leurs relations et les responsabilités associées à chaque module. L'architecture repose sur trois ensembles principaux : la configuration, la supervision SNMP, et la journalisation (logs).



4.1. Module de configuration

Le module de configuration est géré par la classe *ConfigManager*, responsable du chargement, de la sauvegarde et de la gestion des équipements supervisés. Les équipements sont représentés par la classe *Device*, qui contient les informations essentielles à leur identification et à leur supervision :

- nom, adresse IP, type d'appareil, adresse MAC et marque
- liste d'OID associés pour la surveillance des différents paramètres.

Chaque *Device* peut être associé à un *ModeleSupervision*, qui définit les paramètres surveillés via des objets de type *ParametreSurveille* (par exemple CPU, mémoire, trafic réseau). Ces paramètres sont eux-mêmes liés à des objets *OID*, qui définissent les identifiants SNMP à interroger, et à des objets *Seuil*, utilisés pour déterminer les conditions d'alerte.

Le *ConfigManager* interagit avec un fichier de configuration (au format JSON) contenant la description des équipements. Ce fichier est lu au démarrage du programme afin de créer dynamiquement les objets *Device*. Grâce à l'interface web, l'utilisateur (selon ses droits) peut ajouter, modifier ou supprimer des équipements directement, et ces changements sont ensuite sauvegardés dans le fichier JSON.

4.2. Module de supervision

Le module de supervision est représenté par la classe *MoniteurSNMP*, responsable du processus de requêtage SNMP des équipements. Chaque moniteur est associé à un objet *Device* et interroge périodiquement ses OID pour récupérer les valeurs de performance.

Ces données sont stockées dans la classe *DataDevice*, qui contient :

- un horodatage (timestamp).
- un dictionnaire (valeurs) associant chaque OID à la valeur mesurée.

Le *MoniteurSNMP* est également capable de détecter les dépassements de seuils grâce à la classe *Alerte*, qui relie un *Device*, un *OID* et une valeur critique. Lorsqu'une alerte est générée, elle peut être transmise à l'*InterfaceUtilisateur*, qui se charge de l'afficher sous forme graphique ou textuelle (tableaux, graphiques de performances, alertes colorées).

Enfin, la classe *InterfaceUtilisateur* centralise l'interaction avec l'utilisateur final. Elle permet de visualiser l'état général du parc supervisé, les alertes en cours et l'évolution des indicateurs sous forme de graphiques.

4.3. Module des logs

Le module de logs, non représenté comme une seule classe dédiée dans le diagramme, repose sur l'enregistrement des événements dans la base de données ou dans des fichiers texte. Chaque *Alerte* ou événement généré par le *MoniteurSNMP* est consigné avec un horodatage, le nom du *Device* concerné et le message d'erreur ou d'information associé.

Ce module permet ainsi de :

- conserver un historique complet des événements pour chaque machine ;
- faciliter la traçabilité des incidents ;
- fournir des données exploitables pour l'analyse des performances ou le diagnostic de pannes.

Les logs seront consultables depuis l'interface web, avec un filtrage par appareil, type d'erreur ou période temporelle.

4.4. Conclusion du diagramme de classe

Ce diagramme illustre une architecture capable de s'adapter, où chaque composant (configuration, supervision, journalisation) est isolé et interagit via des classes clairement définies. Cette approche facilite la maintenance, les tests unitaires et l'intégration de nouvelles fonctionnalités, comme la personnalisation des modèles de supervision ou la génération automatique de rapports d'activité.

5. Technologies utilisées

Pour le développement de ce projet, nous avons choisi d'utiliser le langage **Python**, adapté à la fois pour la supervision SNMP et pour la gestion des fichiers de configuration. Il est utilisé dans les trois modules principaux : configuration, surveillance et journalisation.

5.1 Technologies logicielles

- **Python** : langage principal, choisi pour sa simplicité, ses bibliothèques réseau et sa compatibilité multiplateforme.
 - *pysnmp* pour les échanges SNMP (interrogation et collecte des OID)
 - *json* pour la gestion du fichier de configuration
 - *logging* pour l'écriture des logs systèmes

5.2 Technologies web

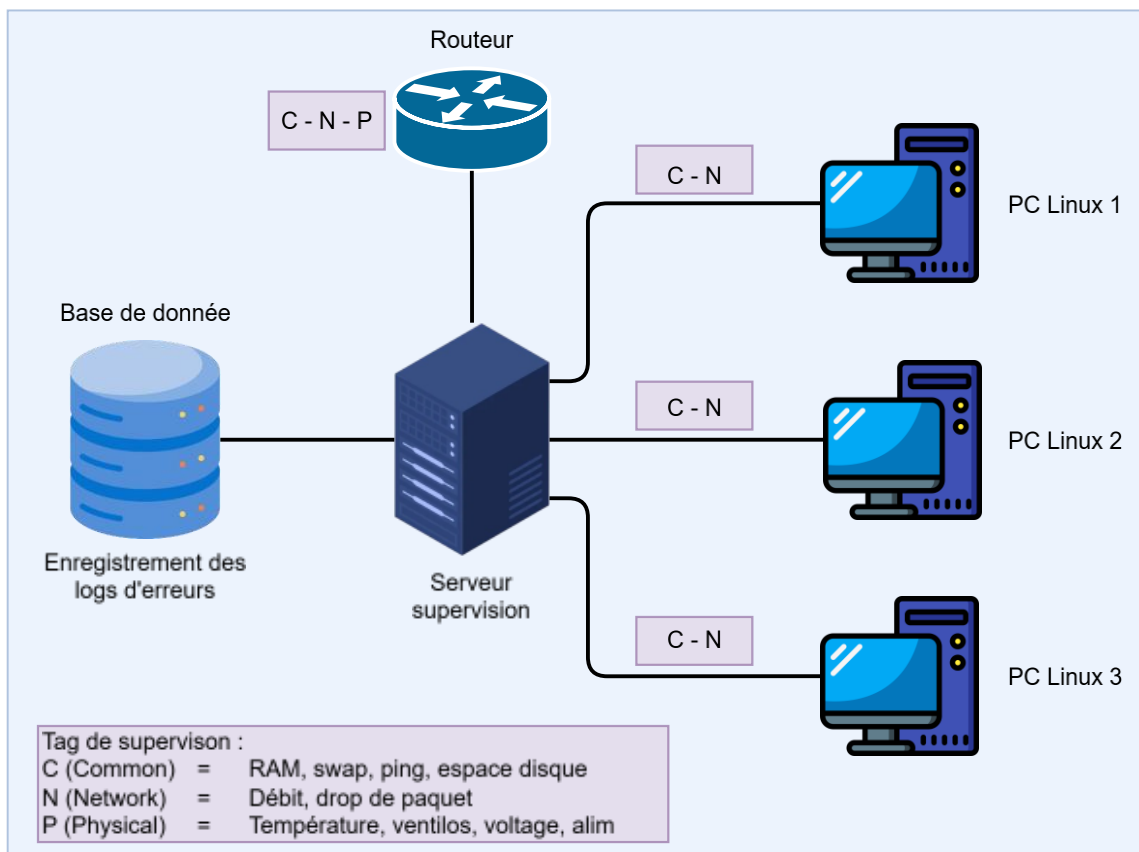
- **HTML / CSS** : pour la conception de l'interface utilisateur. L'objectif est d'offrir une interface simple et intuitive permettant la consultation des données de supervision et la configuration des appareils. L'interface différencie les droits d'accès entre administrateur et utilisateur standard.
- **Flask** : micro-framework web Python utilisé pour créer l'interface utilisateur et gérer la communication entre la partie back-end (modules Python) et le front-end (HTML/CSS). Il permet de lancer un serveur web local, d'afficher les données supervisées en temps réel et d'interagir avec le fichier de configuration.

5.3 Stockage des données

- Les informations relatives aux machines supervisées sont sauvegardées dans un fichier JSON, permettant une configuration flexible sans base de données. Ce format facilite l'ajout ou la suppression de nouveaux équipements via l'interface web et la lecture automatique par le module de configuration.

6. Schéma réseau

Le schéma ci-dessous illustre l'architecture réseau qui sera mis en place pour le projet. Il présente les différents éléments impliqués et les échanges entre eux.



Le serveur de supervision exécute les modules Python (configuration, surveillance et logs) et héberge l'application Flask permettant d'accéder à l'interface web. Les équipements réseau supervisés (commutateurs, routeurs, serveurs) répondent aux requêtes SNMP envoyées par le serveur. Les utilisateurs (administrateurs et utilisateurs standards) se connectent via un navigateur web pour consulter les données de supervision ou modifier la configuration.

Les principaux flux sont les suivants :

- SNMP** : échanges entre le serveur de supervision et les équipements pour la collecte des métriques (CPU, RAM, trafic, etc.) ;
- HTTP** : échanges entre l'utilisateur et le serveur Flask pour la consultation et la gestion du parc supervisé ;
- Fichier JSON** : stockage local des configurations des machines (ajout, suppression ou modification d'équipements).

7. Gantt & avancement du projet

Cette partie présente le plan de développement établi pour le projet ainsi que son état d'avancement actuel. Chaque tâche a été identifiée et attribuée à un membre du groupe, avec un suivi de son état (terminée, en cours, bloquée ou à faire).

Le développement du projet se déroule selon plusieurs grandes étapes :

- Analyse et préparation (lecture du sujet, définition des rôles et des use cases)
- Conception (diagrammes UML, format JSON, schéma réseau)
- Implémentation (modules de configuration, surveillance et logs, interface web, base de données, machines virtuelles)
- Tests et validation (prévue pour la fin du développement)

Le tableau ci-dessous résume l'ensemble des tâches prévues et leur état d'avancement :

Tr	Tâche	Propriétaire	État	Date de début	Date de fin	Tr	Notes
	Lire et comprendre le sujet	Alexandre LAKOMY clement3970@gmail.com	Terminé	15/09/2025	15/09/2025		Analyse & préparation
	Identifier les rôles (Admin / Utilisateur)	Alexandre LAKOMY clement3970@gmail.com	Terminé	15/09/2025	15/09/2025		Analyse & préparation
	Définir les cas d'usage (Use Cases)	Alexandre LAKOMY clement3970@gmail.com	Terminé	15/09/2025	15/09/2025		Analyse & préparation
	Définir le cadre de la supervision	Alexandre LAKOMY clement3970@gmail.com	Terminé	15/09/2025	15/09/2025		Analyse & préparation
	Définir le format du fichier (JSON)	Alexandre LAKOMY clement3970@gmail.com	Terminé	15/09/2025	15/09/2025		Analyse & préparation
	Concevoir le diagramme de Gantt	Alexandre LAKOMY clement3970@gmail.com	Terminé	17/09/2025	25/09/2025		Conception
	Concevoir le schéma réseau	Alexandre LAKOMY	Terminé	17/09/2025	17/09/2025		Conception
	Concevoir le diagramme de classes	clement3970@gmail.com	Terminé	17/09/2025	25/09/2025		Conception
	Concevoir le diagramme des Use Cases	Alexandre LAKOMY	Terminé	17/09/2025	25/09/2025		Conception
	Création de la base de données	clement3970@gmail.com	Terminé	25/09/2025	dd/mm/yyyy		Base de Données
	Ajout des tables	clement3970@gmail.com	Terminé	25/09/2025	dd/mm/yyyy		Base de Données
	Création site web basique	Alexandre LAKOMY	Terminé	25/09/2025	dd/mm/yyyy		Interface Web
	Faire l'interconnexion Web/BDD	Alexandre LAKOMY clement3970@gmail.com	En cours	25/09/2025	dd/mm/yyyy		Interface Web & Base de Données
	Récupération données NAS - Test	Alexandre LAKOMY	Terminé	09/10/2025	dd/mm/yyyy		Test de récupération OID
	Définir chaque OIDs nécessaire	clement3970@gmail.com Alexandre LAKOMY	En cours	09/10/2025	dd/mm/yyyy		Notes
	Ajouter/Supprimer/Modifier machine	À Nom	À faire	09/10/2025	dd/mm/yyyy		Développement du module Supervision
	Gérer les OIDs	À Nom	À faire	09/10/2025	dd/mm/yyyy		Développement du module Supervision
	Seuil	À Nom	À faire	09/10/2025	dd/mm/yyyy		Développement du module Supervision
	Template	À Nom	À faire	09/10/2025	dd/mm/yyyy		Développement du module Supervision
	Création des machines virtuelles	À Nom	À faire	16/10/2025	dd/mm/yyyy		Machines Virtuelles
	Configuration des machines virtuelles	À Nom	À faire	16/10/2025	dd/mm/yyyy		Machines Virtuelles
	Implémenter la journalisation	À Nom	À faire	24/10/2025	dd/mm/yyyy		Développement du module Logs & Alertes
	Implémenter la détection de seuils	À Nom	À faire	24/10/2025	dd/mm/yyyy		Développement du module Logs & Alertes
	Implémenter la détection d'alertes	À Nom	À faire	24/10/2025	dd/mm/yyyy		Développement du module Logs & Alertes
	Page de consultation des machines supervisées	À Nom	À faire	24/10/2025	dd/mm/yyyy		Interface Utilisateur
	Page de consultation des alertes	À Nom	À faire	13/11/2025	dd/mm/yyyy		Interface Utilisateur
	Page de consultation des logs	À Nom	À faire	13/11/2025	dd/mm/yyyy		Interface Utilisateur
	Page de consultation des historiques	À Nom	À faire	13/11/2025	dd/mm/yyyy		Interface Utilisateur
	Test à effectuer	À Nom	À faire	27/11/2025	dd/mm/yyyy		Tests

8. Bilan de mi-projet

À ce stade du projet, la majorité des tâches de conception ont été réalisées, notamment la définition du cadre de supervision, la modélisation des classes, et la mise en place de la structure logicielle. Le développement des modules principaux (configuration, supervision et logs) a débuté conformément au planning prévu. Il est possible qu'il y ait quelques changements de conception durant la suite du projet.

Durant les dernières séances, nous avons franchi des étapes importantes en créant la base de données et ses tables principales, destinées à stocker les informations de supervision. Nous avons également développé la structure du site web avec Flask, en intégrant les pages essentielles (accueil, configuration, logs, dashboard, etc.). Enfin, nous avons établi la connexion entre le site et la couche SNMP, permettant de récupérer et d'afficher en temps réel les informations du NAS et du switch directement sur l'interface web.

Quelques fonctionnalités, telles que la détection des alertes ou la gestion avancée des seuils, restent encore à implémenter. Cependant, le projet avance selon le plan initial et dispose désormais d'une première version fonctionnelle de l'outil de supervision réseau.