



Table des matières

[1. Introduction 2](#_heading=h.gjdgxs)

[2. Cahier des charges 4](#_heading=h.30j0zll)

[3. Les différents composants du projet 6](#_heading=h.n45hsi5xexz1)

[Les différents éléments composant le projet : 6](#_heading=h.633phlab8nz7)

[4. Page Login, mise en place du système de token et création d’un script d’automatisation 8](#_heading=h.gbj1em5qij3j)

[a. Qu'est-ce qu'un Token ? 8](#_heading=h.deh11s6fj5gk)

[b. Types de Tokens 8](#_heading=h.j89ih75w7nuw)

[c. Sécurité des Tokens 8](#_heading=h.oqjep0nk2zye)

[Fonctionnalités du Code gérant les Tokens : 9](#_heading=h.oollya7i6dvd)

[Fonctionnalités des Servlets : 10](#_heading=h.87hxys6kxcdl)

[Voici les différentes fonctionnalités du Script : 11](#_heading=h.f161be2dujnu)

[5. Page de gestion des équipements et gestion des utilisateurs 13](#_heading=h.cnd0q1we6k6l)

[Fonctionnalités du Code gérant les équipements : 13](#_heading=h.w8cal0bicmnq)

[6. Page des tickets 15](#_heading=h.j7j75tvqeuy6)

[7. Difficultés rencontrées 15](#_heading=h.fstvttf20g4w)

[8. Choisir c’est renoncer 16](#_heading=h.iu11wpryf62e)

# Introduction

La Gestion de Maintenance Assistée par Ordinateur (G.M.A.O) est un outil essentiel pour optimiser la gestion des opérations de maintenance au sein d'une entreprise, en particulier pour une PME et ETI / GE. Pour les techniciens réseaux, l'implantation d'une GMAO apporte de nombreux avantages, en facilitant la supervision et l'organisation de la maintenance des infrastructures informatiques et réseau.

Voici la liste des fonctions d’une GMAO conçues pour les techniciens réseaux dans une PME :

* Gestion centralisée des équipements : La GMAO permet de recenser et suivre l'état de l'ensemble des équipements réseau (routeurs, switches, serveurs, etc.). Chaque appareil est répertorié avec ses spécifications, son historique de maintenance et ses performances.
* Planification de la maintenance préventive : Grâce à une GMAO, il est possible de programmer des interventions régulières pour éviter les pannes imprévues. Cela permet d'établir un calendrier précis de maintenance préventive, en fonction de la durée de vie des équipements et des recommandations des fabricants.
* Réduction des temps d'arrêt (downtime) : En suivant et en anticipant les besoins de maintenance via la GMAO, les techniciens réseaux peuvent prévenir les pannes, ce qui diminue les interruptions de service. Moins de temps d'arrêt se traduit par une productivité accrue et une meilleure satisfaction client.
* Suivi des interventions : Chaque intervention technique, qu'elle soit corrective ou préventive, est enregistrée dans la GMAO. Cela permet de garder une trace des actions effectuées, facilitant ainsi les audits, et permettant une analyse rétrospective en cas de dysfonctionnement récurrent.
* Gestion des stocks et des pièces détachées : La GMAO permet de suivre les niveaux de stock des pièces de rechange (câbles, composants de réseau, etc.). Les techniciens peuvent rapidement savoir ce qui est disponible et éviter les retards dus à une pénurie de matériel.
* Amélioration de la communication et collaboration : Avec un système GMAO, les techniciens et les autres membres de l'équipe (comme les gestionnaires ou les autres départements) peuvent accéder en temps réel aux informations sur l'état des équipements et les interventions en cours. Cela améliore la coordination et permet une meilleure prise de décision.
* Analyse des coûts et optimisation des ressources : La GMAO permet d’analyser les coûts de maintenance et de repérer les sources de gaspillage ou d’inefficacité. Ainsi, les techniciens peuvent mieux allouer les ressources et optimiser l'utilisation des équipements réseau.

Dans une PME, la GMAO devient un outil stratégique pour les techniciens réseaux, leur permettant de mieux gérer les infrastructures tout en réduisant les coûts et en optimisant les performances. En automatisant et en centralisant la gestion des tâches de maintenance, elle permet aux équipes techniques de se concentrer sur des missions à plus forte valeur ajoutée, tout en garantissant la fiabilité des réseaux et systèmes.

# Cahier des charges

Pour la création de notre GMAO (Gestion de Maintenance Assistée par Ordinateur) à destination des techniciens réseaux d'une PME, nous avons élaboré un cahier des charges adapté. Ce dernier répond aux exigences définies lors de l’introduction du projet. L’objectif est de développer un projet réalisable en environ 50 heures de travail.

Objectifs du projet :

1. Mettre en place une GMAO sous forme de site web accessible uniquement en local au sein de l’entreprise.
2. Utiliser des technologies simples, comme HTML, pour l'interface utilisateur.
3. Sécuriser l'accès au site avec un système d'authentification.
4. Créer une structure adaptée aux besoins des différents profils d'utilisateurs : secrétaire, technicien et administrateur.

**Étapes et contraintes du projet :**

*Accès et sécurité :*

* La GMAO sera un site web local accessible uniquement depuis le réseau interne de l’entreprise.
* Le site inclura un système d'authentification sécurisé, avec des options de déconnexion pour chaque utilisateur.

*Trois profils d'utilisateur seront mis en place :*

* Secrétaire : Accès limité à la page « Demandes techniques » pour soumettre des requêtes concernant des besoins ou problèmes réseau spécifiques.
* Techniciens : Accès à plusieurs pages, dont :
* Une page dédiée à l'inventaire où ils pourront ajouter et gérer les équipements réseaux (ordinateurs, switchs, routeurs, points d’accès, câbles, etc.).
* Possibilité de réaliser des recherches avancées et de trier les équipements selon divers critères.
* Option d’export des équipements sélectionnés au format Excel, avec la possibilité d’imprimer des listes.
* Une page pour consulter et traiter les demandes des autres départements, concernant généralement des problèmes mineurs ou des pannes.
* Une page avec un planning des maintenances à venir, incluant la planification des réparations et interventions sur les équipements.
* Administrateur : Accès complet avec des droits étendus, lui permettant de modifier ou supprimer des équipements, interventions, demandes, et autres données du système.

*Base de données :*

* Le système devra être connecté à une base de données (SQL ou PostgreSQL) hébergée localement sur un serveur interne, comme le site lui-même.
* La base de données devra être sécurisée et accessible uniquement par les administrateurs et utilisateurs autorisés.

*Contraintes techniques :*

* Langage de programmation : Le site sera développé en HTML pour l'interface utilisateur. Un langage côté serveur comme PHP ou Python peut être utilisé pour l’interaction avec la base de données.
* Système d'authentification : L'authentification devra être sécurisée (via HTTPS et mots de passe chiffrés).
* Sécurité : Les accès aux données seront restreints selon les profils utilisateurs, et des sauvegardes régulières devront être mises en place pour assurer la continuité de service en cas de panne.

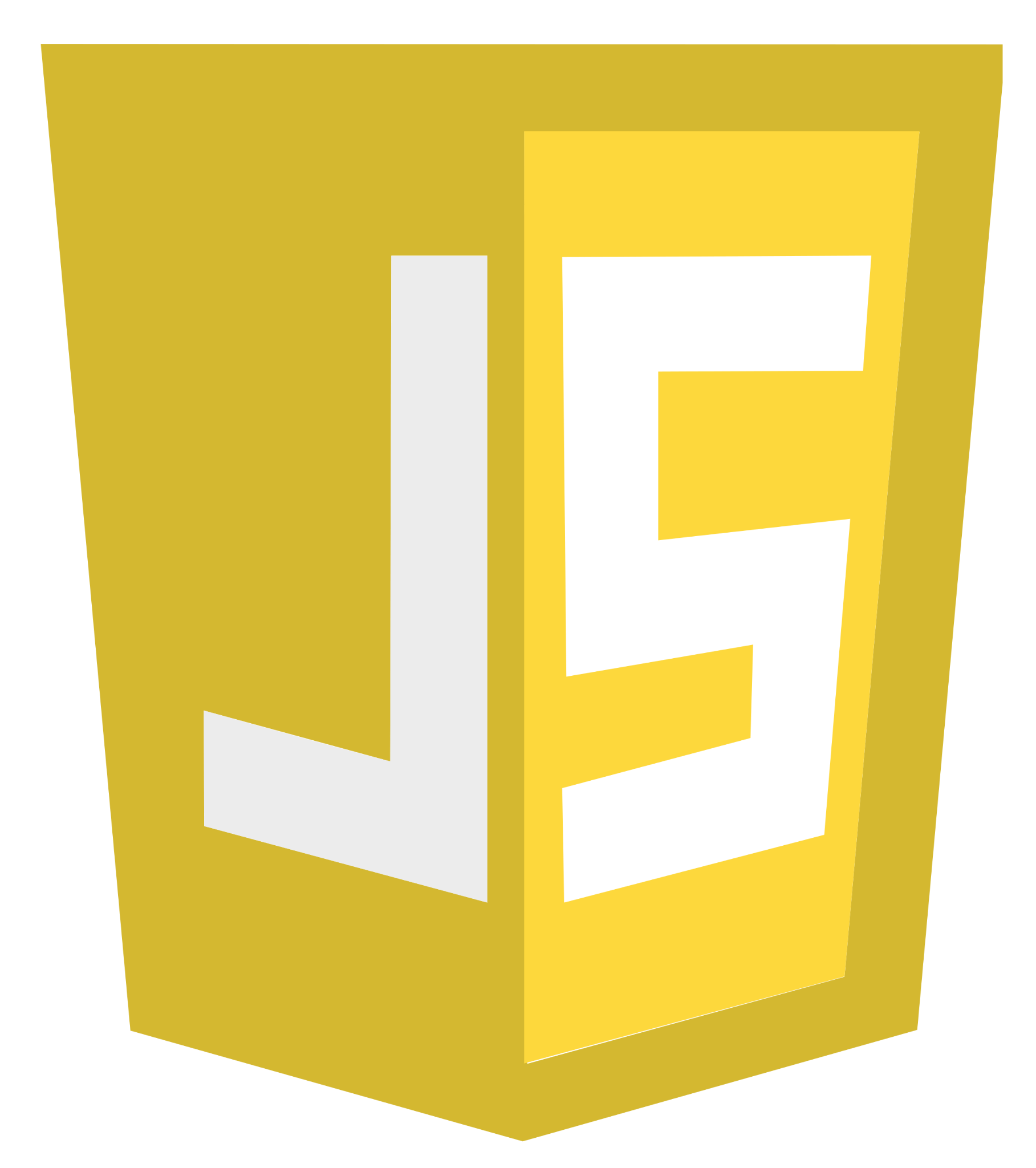
Ce cahier des charges définit les grandes lignes du projet de GMAO pour les techniciens réseaux d'une PME, en se concentrant sur la sécurité, l'organisation de l'inventaire, et la gestion des demandes techniques pour optimiser la gestion des équipements et interventions.

# 3. Les différents composants du projet

### **Les différents éléments composant le projet :**

| **Outil/Technologie** | **Description** | **Utilisation dans le projet** |
| --- | --- | --- |
| **PostgreSQL** | Système de gestion de base de données relationnelle et objet, open source sous licence BSD, reconnu pour sa richesse fonctionnelle et sa conformité aux standards SQL. | Stockage des données pour l'application, gestion des attributs des équipements. |
| **Apache Tomcat** | Serveur d’applications web open source conçu pour exécuter des applications Java, développé par la fondation Apache. | Hébergement et déploiement des servlets Java. |
| **NetBeans** | Environnement de développement intégré (IDE) open source, supportant Java et d'autres langages (C, C++, JavaScript, etc.), utilisé principalement pour gérer les servlets Java dans le cadre de la GMAO. | Développement de la partie servlets Java de la gestion de maintenance assistée par ordinateur. |
| **Ubuntu** | Distribution Linux open source basée sur le noyau Linux, gratuite et modifiable, lancée en 2004 par Canonical Ltd, reposant sur Debian. | Hébergement des services Apache Tomcat et de la base de données PostgreSQL. |
| **Visual Studio** | IDE créée par Microsoft, supportant plusieurs langages (C#, VB.NET, C++, etc.), permettant le développement d'applications sur diverses plateformes. | Utilisé avec GitHub pour intégrer les éléments du site, les servlets et la base de données PostgreSQL dans un environnement collaboratif. |
| **GitHub** | Plateforme de gestion de versions et de collaboration pour le développement logiciel, facilitant le suivi des modifications de code et la coopération entre les développeurs. | Assure l'intégration entre Visual Studio, les servlets Java et PostgreSQL, favorisant le travail collaboratif et le partage des connaissances. |

***Les langages utilisées lors de ce projet :***



# 4. Page Login, mise en place du système de token et création d’un script d’automatisation

Pour garantir la sécurité de notre Gestion de Maintenance Assistée par Ordinateur (GMAO) et structurer les différents utilisateurs qui testeront notre programme, nous avons décidé de mettre en place un système d'authentification.

Nous avons opté pour une base HTTPS afin de sécuriser les communications et de mettre en œuvre une méthode d'authentification par tokens. Cette approche permettra de protéger les données échangées tout en assurant une identification fiable des utilisateurs.

### **a. Qu'est-ce qu'un Token ?**

Un **token** est une chaîne de caractères générée pour représenter une session ou une information d'authentification. Il est utilisé pour sécuriser les communications entre un client (navigateur) et un serveur.

### **b. Types de Tokens**

Il existe plusieurs types de tokens, dont :

* **Tokens d'authentification** : Utilisés pour vérifier l'identité d'un utilisateur. Par exemple, un token JWT (JSON Web Token) contient des informations sur l'utilisateur et peut être utilisé pour authentifier les requêtes auprès d'un serveur.
* **CSRF Tokens (Cross-Site Request Forgery)** : Utilisés pour protéger les formulaires contre les attaques CSRF. Un token CSRF est généré par le serveur et inclus dans les formulaires HTML. Il doit être soumis avec le formulaire pour prouver que la requête provient d'un utilisateur légitime.

### **c. Sécurité des Tokens**

Il est crucial de protéger les tokens pour éviter les attaques. Les bonnes pratiques incluent :

* **Utilisation de HTTPS** : Pour sécuriser la transmission des tokens.
* **Expiration des tokens** : Pour limiter la durée pendant laquelle un token est valide.
* **Rafraîchissement des tokens** : En utilisant un mécanisme pour générer de nouveaux tokens avant l'expiration des anciens.

Les tokens jouent un rôle crucial dans la sécurisation de la GMAO. Ils facilitent l'authentification des utilisateurs et permettent de suivre les connexions ainsi que les échanges entre les utilisateurs et le serveur, garantissant ainsi une communication sécurisée et fiable.

D’autre part, pour que les tokens prennent effet, il a fallu créer un code dans le script des pages HTML et implémenter de nouvelles méthodes dans NetBeans, en particulier dans la DAO.

### **Fonctionnalités du Code gérant les Tokens :**

1. **Reload sur Déconnexion** :
   * La fonction window.onunload est configurée pour appeler reload() lorsque la page est quittée. Cela permet potentiellement de rafraîchir la page à chaque fois qu'un utilisateur se déconnecte ou quitte la page (il faudrait définir la fonction reload()).
2. **Récupération de l'Adresse IP du Serveur** :
   * L'adresse IP du serveur est récupérée à partir de l'URL de la page actuelle. Cela est utile pour accéder au serveur Apache et, par extension, au serveur Tomcat.
3. **Gestion des Données de Session** :
   * Les données de session (tokenGMAO, userLogin, et userRole) sont extraites de sessionStorage. Cela permet de conserver l'état de la session de l'utilisateur.
4. **Vérification du Token** :
   * Si le tokenGMAO est défini, une requête POST est envoyée au serveur pour vérifier la validité du token. La réponse est ensuite traitée par la fonction TokenCheck.
   * Si le tokenGMAO n'est pas présent, l'utilisateur est redirigé vers la page de connexion (login.html).
5. **Traitement de la Réponse de Vérification** :
   * La fonction TokenCheck traite la réponse JSON retournée par le serveur. En fonction du rôle de l'utilisateur (Admin, Technicien, Utilisateur), il est redirigé vers la page appropriée.
   * Si le rôle n'est pas reconnu ou si le token est invalide, l'utilisateur est également redirigé vers la page de connexion.

### **Fonctionnalités des Servlets :**

Dans un second temps, nous avons jugé nécessaire de créer un script permettant la reconstitution et l’automatisation des données dans NetBeans. Cette démarche vise à optimiser le développement en faisant gagner un temps précieux et à faciliter l'accès à la GMAO depuis tous les postes étudiants.

### **Voici les différentes fonctionnalités du Script :**

1. **Vérification et Démarrage de PostgreSQL** :
   * Le script commence par vérifier l'état du service PostgreSQL. S'il est inactif, il le démarre.
2. **Gestion des Fichiers de l'Application** :
   * Il vide le répertoire de l'application web Apache (sous /var/www/gmao) pour s'assurer qu'aucun ancien fichier n'est présent.
   * Il crée un répertoire pour la documentation Javadoc et copie les fichiers nécessaires depuis un répertoire local vers le répertoire Apache.
3. **Vérification et Démarrage d'Apache** :
   * Il vérifie ensuite si le serveur Apache est en cours d'exécution. Si ce n'est pas le cas, il effectue un démarrage.
4. **Reconstruction de la Base de Données** :
   * Le script propose à l'utilisateur de reconstruire la base de données. Si l'utilisateur accepte, il supprime la base de données existante (sae\_52) et peut éventuellement exécuter un script de configuration pour la reconstruire.
5. **Lancement de NetBeans** :
   * Il vérifie si NetBeans est déjà en cours d'exécution. Si ce n'est pas le cas, il demande à l'utilisateur s'il souhaite le lancer et, en cas d'acceptation, ouvre NetBeans dans un nouvel onglet de terminal.
6. **Rapport d'Exécution** :
   * Enfin, le script affiche l'état des services PostgreSQL et Apache, ainsi que l'état de NetBeans. Si la base de données a été reconstruite, il informe l'utilisateur que l'opération a été réalisée avec succès.
7. **Accès à la Documentation** :
   * Il indique également l'emplacement de la documentation Javadoc générée, accessible via une URL.

# Page de gestion des équipements et gestion des utilisateurs

Cette page est principalement destinée à l'administrateur, qui doit avoir la capacité d'ajouter et de supprimer divers éléments, tels que des ordinateurs, des routeurs, des switches et des câbles, au sein d'une PME.

Pour commencer, nous avons conçu une page HTML qui inclut plusieurs boutons et un tableau demandant les caractéristiques à remplir, ainsi qu'une liste permettant d'afficher les équipements enregistrés. Pour mettre en place ce système, nous avons d'abord créé des tables dans PostgreSQL (pc, switch, router et cable) qui permettront de stocker les différents attributs de chaque équipement, tels que l'adresse MAC, le VLAN, le numéro de série, le nom, etc.

Ensuite, nous avons développé un code HTML qui permet de détecter la méthode à appeler en fonction de l'objet sélectionné. Ainsi, le code interagit de la manière suivante :

### **Fonctionnalités du Code gérant les équipements :**

1. **Écoute de l'Événement de Clic** :
   * Un écouteur d'événements est ajouté au bouton submitButton. Lorsqu'il est cliqué, il empêche le comportement par défaut du formulaire (rechargement de la page) et détermine le type d'équipement sélectionné dans un sélecteur (equipmentTypeSelect).
2. **Redirection selon le Type d'Équipement** :
   * En fonction du type d'équipement sélectionné (PC, routeur, switch, ou câble), la fonction correspondante est appelée (addPC, addRouter, etc.).
3. **Ajout d'un PC (identique pour les switchs, routeurs et câbles)**  :
   * Dans la fonction addPC, les valeurs des champs du formulaire sont récupérées.
   * Une requête fetch est ensuite effectuée pour envoyer les données au serveur via une requête POST à l'URL spécifiée.
4. **Vérification de la Réponse** :
   * La réponse du serveur est traitée. Si le résultat indique que le nom de l'équipement existe déjà, un message d'alerte est affiché. Sinon, vous pouvez ajouter d'autres actions (non spécifiées dans votre code).
5. **Réinitialisation du Formulaire** :
   * Le formulaire est réinitialisé après l'envoi des données, quelle que soit la réponse.

Maintenant, au niveau de NetBeans, nous avons dû réaliser différents servlets pour coïncider avec le code précédent.

# Page des tickets

# Difficultés rencontrées

# Choisir c’est renoncer