DEPARTEMENT INFORMATIQUE - IUT 2 GRENOBLE

Année Universitaire 2024-2025

MÉMOIRE DE STAGE

MÉMOIRE DE STAGE

Compagnie de Chauffage Intercommunale de l'Agglomération de Grenoble (CCIAG)

Présenté par

Jarod Tivollier

Jury

IUT : Mme. Montanvert Annick

IUT : Mme. Laget-Kamel Sarah

Société : Mr. Nicolas Thomas

Déclaration de respect des droits d'auteurs

Par la présente, je déclare être le seul auteur de ce rapport et assure qu'aucune autre ressource que celles indiquées n'ont été utilisées pour la réalisation de ce travail. Tout emprunt (citation ou référence) littéral ou non à des documents publiés ou inédits est référencé comme tel et tout usage à un outil doté d'IA a été mentionné et sera de ma responsabilité.

Je suis informé qu'en cas de flagrant délit de fraude, les sanctions prévues dans le règlement des études en cas de fraude aux examens par application du décret 92-657 du 13 juillet 1992 peuvent s'appliquer. Elles seront décidées par la commission disciplinaire de l'UGA.

A Grenoble Le 04/06/2025 Signature Jarod Tivollier

Sommaire

Table des figures

I. INTRODUCTION

Ce mémoire de stage est réalisé dans le cadre d'un stage en entreprise obligatoire à réaliser en 2ème année de BUT informatique à Grenoble. Ce stage a été réalisé au sein de l'entreprise CCIAG* (Compagnie de Chauffage Intercommunale de l'Agglomération de Grenoble). Au sein de ce mémoire de stage, vous découvrirez la mission qui m'a été confiée par l'entreprise. Cette mission-là voici, c'était « Évolution et Amélioration de GLPI* (Gestionnaire Libre de Parc Informatique) ». Cette mission a divers objectifs expliqués et détaillés au cours de ce mémoire de stage. Le mémoire de stage est organisé de la façon suivante :

- Une introduction
- Une présentation de l'entreprise d'accueil
- Une présentation du stage avec ses objectifs
- La conception et réalisation de cette mission
- Une conclusion
- Un glossaire
- Des annexes

Pour finir, ce mémoire de stage a été réalisé avant que je puisse finir mon projet. Il y a donc certaines parties que j'explique qui ne sont pas finies.

II. PRESENTATION DU CONTEXTE PROFESSIONNEL

II.1 Présentation de l'entreprise

La Compagnie de Chauffage Intercommunale de l'Agglomération de Grenoble ou CCIAG est l'entreprise qui s'occupe du réseau de chaleur de la métropole de Grenoble. C'est le deuxième plus grand réseau de chaleur en France, le premier étant la CPCU (Compagnie Parisienne de Chauffage Urbain). Créée en 1960, c'est une SEML* (Société d'Économie Mixte Locale) qui est détenue majoritairement par la commune de Grenoble. La CCIAG possède 2 principales activités :

La première, c'est la production et la distribution de chaleur. Le réseau de chaleur de la CCIAG s'étend sur plus de 170km avec un total d'environ 100 000 équivalent logements* chauffés. Pour chauffer toutes ces habitations, la CCIAG compte 4 chaufferies

- La chaufferie de La Poterne
- La chaufferie de Villeneuve
- La chaufferie Biomax
- La chaufferie Athanor

Au cours de l'année, la plupart du temps, les chaufferies ne fonctionnent pas toutes en même temps en raison de diverses lois et réglementations sur l'environnement et aussi qu'en été les habitations n'ont pas besoin de chauffage. Sauf Athanor qui reste en fonctionnement toute l'année en raison de son combustible qui est les déchets ménagers.

La Poterne :

Cette chaufferie située au 30 chemin de la Poterne à Grenoble, mise en service pour la première fois en 1992, utilise comme combustibles du bois, de la farine d'animal, du charbon, de l'huile, du gaz et enfin du fioul. Avant d'avoir reçu sa première rénovation, après cette rénovation, la chaufferie n'utilise plus que du bois, de la farine d'animal et du charbon. En 2026 l'entreprise prévoit d'alimenter la chaufferie exclusivement avec du bois aujourd'hui, elle utilise toujours du charbon pour fonctionner. Cette chaufferie produit aussi de l'éléctricité.

Villeneuve :

Cette chaufferie située à Villeneuve a été mise en service pour la première fois en 1970. La chaufferie a reçu sa première rénovation en 1982 qui lui a permis de pouvoir produire de la chaleur à partir de bois ce qui n'était pas possible avant elle n'utilisait que du charbon, du fioul et du gaz comme combustible et cela a permis d'inscrire cette chaufferie sur une démarche de 100% d'utilisation d'énergie renouvelable. En 2021, elle reçoit encore une rénovation permettant encore de limiter l'utilisation de combustible non renouvelable.

Biomax :

Cette chaufferie située sur la Presqu'Ile de Grenoble, mise en service pour la première fois en 2020, utilise uniquement du bois pour fonctionner et chauffe entre 15000 et 20000 équivalents logements. Mais cette chaufferie produit aussi de l'électricité, dont une partie sert à faire fonctionner la chaufferie et l'autre partie sert pour alimenter environ 10 000 équivalents logements.

Athanor :

Cette UIVE (Unité d'Incinération et de Valorisation Énergétique) est située à La Tronche, qui a été mise en service autour des années 1972 et qui jusqu'en 2025 a reçu des rénovations, fonctionne uniquement en brûlant des déchets ménagers. Elle dispose de 3 fours avec une consommation de 8 tonnes/heure de déchets ménagers au total. Elle est la seule des 4 chaufferies à tourner 365 jours/an en raison de son combustible qui est les déchets ménagers produit tout au long de l'année. Cette chaufferie produit aussi de l'électricité.

Figure 1 : Chaufferie La Poterne

Figure 2 : Chaufferie Athanor

La deuxième activité est l'entretien et la performance énergétiques des bâtiments collectifs. La CCIAG assure la maintenance et l'entretien de ses installations de chauffage situées dans les bâtiments raccordés à leur réseau. Leur réseau de chaleur s'étend sur tout Grenoble, avec une partie du campus UGA. Le raccordement complet de l'UGA sur le réseau de la CCIAG est en cours et devrait finir d'ici quelques années, ainsi qu'un futur projet d'étendre le réseau vers la ville de Seyssinet-Pariset et Fontaine en faisant traverser les tuyaux sous le Drac.

La CCIAG utilise aujourd'hui en 2025 80,9 % d'énergie renouvelable et d'énergie de récupération telle que le bois, la farine d'animal ou encore les ordures ménagères et souhaite atteindre les 100% d'ici 2033, ce qui permettrait de faire une réduction de 12000 tonnes de CO2 par an.

La CCIAG est composée de divers services que ce soit l'administration, l'économie ou encore le service de communication interne et externe, mais elle est aussi composée d'un service numérique. C'est dans ce service que j'ai fait mon stage. Dans ce service, on retrouve un pôle Soluciel* composé de deux personnes qui s'occupent des problèmes liés principalement au logiciel et aux bases de données, Administration Réseau composé de 2 personnes qui s'occupent des problèmes liés au réseau de l'entreprise et enfin le Support composé de deux personnes qui s'occupent un peu de tout mais surtout de l'équipements informatiques de l'entreprise avec leur configuration et installation de logiciels mais qui ne sont pas les mêmes car le Support est géré par une entreprise externe. C'est avec l'aide de tous ces pôles que j'ai pu réaliser ma mission de stage.

II.2 Présentation du stage

Au sein de la CCIAG et plus précisément du service numérique (SN), la mission qui m'a été proposée de faire et que j'ai acceptée était la suivante : "Évolution et Amélioration de GLPI"". Les objectifs de cette mission sont :

- Mettre à jour GLPI (Gestionnaire Libre de Parc Informatique)
- Rajouter une gestion des stocks pour l'équipement informatique non pris en charge par GLPI
- Rajouter un cycle de vie pour l'équipement informatique

Le 3ème objectif c'est rajouter plus tard après une réunion de mi-projet avec le client qui me demande de rajouter un cycle de vie.

Ce projet s'est déroulé comme nous l'apprenons au sein de l'IUT, « en mode projet », avec un client qui est le service numérique de la CCIAG qui demande un besoin auquel je peux y répondre, ce qui mène à l'élaboration d'un cahier des charges à présenter au client.

Pour cela j'ai commencé par une analyse de l'existant qui a permis l'élaboration d'un cahier des charges situé en annexe qui présente en détail tout ce qui est attendu par la mission et ce qu'il y a à savoir sur l'environnement de travail ainsi que le planning des tâches. Ce cahier des charges a été modifié sur quelques points à partir de la présentation de mi-projet avec le client pour des raisons qui sont expliquées ultérieurement dans ce mémoire. Lors de ce projet j'ai été assez libre sur ce que je pouvais faire tout en respectant le cahier des charges et j'ai décidé d'axer ce projet sur la maintenance car au sein du service numérique de la CCIAG ils ne sont pas développeurs et possèdent le minimum à savoir sur le développement d'un logiciel ou d'une application web. Ce n'est pas le domaine métier du service numérique qui est plus axé sur l'administration réseau et la gestion de base de données, c'est pour cela que j'ai décidé d'axer mon projet sur la norme ISO 25000 de qualité logicielle qui est la maintenance.

II.3 Planning des taches

Pour ce projet j'ai réparti les différentes tâches dans différentes étapes qui sont décrites dans le cahier des charges, mais pour chaque étape on retrouve toujours une tâche pour les tests et chaque étape correspond à un domaine différent, qui sont les suivantes :

L'étape 1 : Organisation et prise de connaissances : cette partie est principalement axée sur de la gestion de projet avec l'analyse de l'existant et la création d'un cahier des charges.

L'étape 2 : Préparation de l'environnement de travail : cette partie est principalement axée sur du réseau avec la création de VM* (Virtual Machine) et l'utilisation du logiciel Docker*.

L'étape 3 : Accès à la base de données depuis la VM : cette partie est principalement axée sur la gestion de données depuis une base de données où sont hébergées les données de GLPI.

L'étape 4 : Recherche de potentiels plugins* : cette étape est principalement de la recherche de plugins sur la marketplace* de GLPI ainsi que sur GitHub.

L'étape 5 : Ajout des plugins, modification de l'existant et ajout de fonctionnalités : cette étape sera principalement axée sur du développement Web avec la création d'un plugin pour GLPI.

L'étape 6 : Tests utilisateurs : cette étape correspond au test d'interface pour le plugin. Les tests seront proposés à l'équipe du SN (Service Numérique) de la CCIAG pour leur demander un retour sur l'aspect de l'interface et son utilisation.

Ce projet m'a permis de montrer mes compétences acquises tout au long de ma formation au sein du BUT INFO sur les 2 années à travers divers domaines : réseau, développement, base de données, gestion de projet, mathématiques et communication en entreprise.

III. CONCEPTION ET RÉALISATION D'UN SCRIPT DE MISE À JOUR SEMI AUTOMATIQUE DE GLPI

III.1 Conception

Dans un premier temps, pour cette partie de ma mission, il a fallu réfléchir à si oui ou non on continuait d'héberger GLPI ainsi que sa base de données sur Windows. Le résultat découlant de cette tâche a été de changer de système d'exploitation pour Linux pour avoir accès à un logiciel qui s'appelle Docker, pour plusieurs raisons. La raison principale est que ce projet va principalement s'axer sur le critère de maintenance de la norme ISO 25000 et Docker permet de répondre facilement à ce critère de maintenance.

- Docker est un logiciel natif à Linux qui permet la création de conteneurs* qui sont des petites machines virtuelles qui vont accueillir un service*, comme un serveur Apache* pour une application web ou encore un serveur PostgreSQL* pour une base de données. Ces conteneurs sont modulables assez facilement. On peut les remplacer, les supprimer et les créer avec l'aide d'une commande et d'un fichier appelé Dockerfile* qui contient les instructions permettant la création d'un environnement, l'installation de logiciels, l'ajout de fichiers, la modification de fichiers et plein d'autres fonctionnalités encore qui seront exécutées lors de la création du conteneur.
- Docker permet aussi la création de volumes* qui sont des blocs de données qui vont permettre de sauvegarder une partie d'un conteneur indépendamment de lui. Si le conteneur est supprimé et que les données de ce conteneur étaient contenues dans un volume, alors ses données ne seront pas supprimées. Cela permet la persistance des données. Deuxièmement, comme pour les conteneurs, les volumes sont des blocs qu'on peut attacher facilement au conteneur, toujours en une seule commande.
- De plus, une autre raison d'utiliser Docker, c'est que l'utilisation de Docker permet de mettre en avant mes connaissances dues à la formation du BUT Informatique en 2ème année où nous apprenons Docker.
- En ce qui concerne Linux, Docker est natif à Linux donc je n'ai pas le choix si je veux utiliser Docker mais les avantages que Linux a aussi comparé à Windows c'est que Linux utilise moins de ressources (CPU*, GPU*, RAM*) que Windows ce qui est idéal pour une VM qui va accueillir un serveur Web.

Dans un deuxième temps, après avoir réfléchi à quelques technologies que l'on va utiliser pour notre serveur Web, il fallait construire et configurer notre VM Linux. Pour cela, l'entreprise CCIAG possède un réseau de VM créé sur la plateforme VMware Sphere*, c'est l'équivalent de Proxmox que nous voyons au BUT INFO en 2ème année. Leur réseau, on trouve des hyperviseurs. Ce sont des systèmes qui vont s'occuper d'un groupe de VM. Ensuite, il y a des VM de production, des VM qui sont destinées à la production. On va retrouver par exemple la VM Windows de GLPI, et enfin la VM de tests destinés à faire des tests.

J'ai donc créé une VM de test, je l'ai configurée, puis j'ai installé Docker. Après cela, la 2ème étape de cette partie du projet était finie.

Dans un troisième temps, un nouveau temps de réflexion s'impose dans la mesure que le projet s'axe sur le critère de maintenance de la norme ISO 25000. On trouve dans ce critère 4 points que j'ai suivis au cours du projet :

- La stabilité
- L'analysibilité
- La testabilité
- La modifiabilité

Chaque critère est vérifiable selon certains points qui sont :

- La stabilité
  - C'est le fait que le code/script soit stable et ne déclenche pas d'erreur qui pourrait causer des dommages fatals à l'installation. On résout principalement cela avec de la gestion d'erreurs.
- L'analysibilité
  - C'est le fait que le code/script soit lisible, clair et doté de commentaires pour permettre une compréhension plus simple et efficace pour la personne qui aura à le lire.
- La testabilité
  - C'est le fait que le code/script soit testé au préalable pour découvrir les potentielles erreurs ainsi que les corriger.
- La modifiabilité
  - C'est la capacité d'un code/script à être modifiable. Cela passe par diverses solutions comme l'ajout de ce qu'on peut modifier dans le code depuis des variables

Pour respecter cela, j'ai proposé comme solution au service numérique de la CCIAG la création d'un script de mise à jour semi-automatique de GLPI qui a été accepté. Ce script utilise les technologies suivantes :

- Docker
- Bash*

Ce script devra contenir plusieurs instructions permettant :

- La recherche et vérification d'une version de GLPI demandée par l'utilisateur
- La vérification du nombre de plugins utilisés par la version actuelle de GLPI
- La recherche, vérification et compatibilité de ces plugins avec la nouvelle version de GLPI
- L'ajout potentiel de plugin
- La configuration du serveur web au sein du conteneur
- La création du Dockerfile
- L'arrêt et la suppression des conteneurs et des images utilisées par l'ancienne version de GLPI
- Le nettoyage du système Docker
- La création des conteneurs destinés à la nouvelle version de GLPI
- La sécurisation des conteneurs
- Gérer les potentielles erreurs du script
- Respecter le critère de Maintenabilité de la norme ISO 25000 pour ce script

III.2 Réalisation

Pour créer ce script j'utilise différentes connaissances apprises au cours de ma formation de BUT INFO, telles que les appels API* (Application Programming Interface) pour les recherches des plugins qui sont pour la plupart sur les serveurs de GitHub*, ou encore les automates avec les regex* pour la gestion d'erreurs. Étant donné que c'est un script semi-automatique, le script demande plusieurs informations à la personne qui s'occupera de la mise à jour, mais il y a aussi plusieurs informations à savoir au préalable qui sont regroupées dans le « README.md » en annexe

Le script est découpé en 3 sections. La première ce sont les variables qui permettent de modifier le fonctionnement du script, qui permettent de vérifier le sous-critère de la modifiabilité de la Maintenabilité. La 2ème section contient une fonction, cette fonction est celle qui s'occupe d'aller rechercher des plugins sur GitHub depuis l'API GitHub. On retrouvera d'ailleurs dans la première section de token* qui permet l'accès à l'API de GitHub et enfin la 3ème section qui contient le script de mise à jour avec 6 étapes qui sont :

- **Étape 0 : Création du Dockerfile** : dans cette étape, on retrouve la création d'un fichier nommé « DockerfileGLPIcreation » ainsi que diverses instructions qu'on lui rajoute : son image, les dépendances de PHP* (Hypertext Preprocessor), et l'URL* (Uniform Ressource Locator) de l'application web.
- **Étape 1 : Récupération de la dernière version de GLPI et sauvegarde de la clé d'accès* à la marketplace** : dans cette étape, on sauvegarde la clé d'utilisation de la marketplace de GLPI dans un premier temps, puis on demande à l'utilisateur la version de GLPI qu'il souhaite installer et enfin l'ajout d'instructions permettant la configuration du serveur web.
- **Étape 2 : Récupération des différents plugins avant la mise à jour** : Cette partie permet d'informer l'utilisateur des différents plugins qui seront mis à jour. Le script récupère ces informations depuis le dossier plugin de GLPI.
- **Étape 3 : Vérification et récupération des versions les plus récentes des différents plugins** : Dans cette partie, le script exécute la fonction située dans la 2ème section de ce fichier et effectue une recherche pour demander à l'utilisateur quel auteur du plugin souhaite-t-il prendre cette recherche. Lorsque j'ai créé ce script-là, plus précis, je cherche depuis le nom du plugin et si son nom, sa description ou son topic contiennent le mot GLPI. Cette fonction s'exécute autant de fois qu'il y a de plugins ajoutés depuis l'ancien GLPI, puis enfin le script ajoute les instructions permettant à GLPI de pouvoir accéder à ses fichiers.
- **Étape 4 : Destruction et reconstruction des conteneurs Docker** : sur cette partie, le script commence par arrêter et supprimer les conteneurs ainsi que leurs images associées, puis il nettoie le cache* de Docker, puis enfin reconstruit les conteneurs avec le Dockerfile « DockerfileGLPIcreation » que le script aura créé.
- **Étape 5 : Sécurité** : cette étape permet juste qu'on ne puisse pas accéder aux fichiers de GLPI depuis la VM en donnant des droits.

Cette partie du projet aura pris la moitié du temps de mon stage, c'est-à-dire 5 semaines en tout, pour être réalisée. Cette partie du projet se finit avec une première version du script fonctionnel permettant la mise à jour automatique de GLPI ainsi que ses conteneurs, sachant qu'une autre petite partie de cette partie a été faite plus tard, qui est la complétion du README.md ainsi que l'ajout des commentaires dans ce script.

IV. CONCEPTION ET REALISATION D'UN PLUGIN POUR GLPI

IV.1 Conception

Avant cette 2ème partie, j'ai fait une présentation de mon projet au client (le service numérique de la CCIAG) pour leur présenter l'avancement du projet. Au stade de cette présentation, il y a uniquement la première partie du projet qui a été faite. C'est d'ailleurs au cours de cette présentation que j'ai présenté le premier prototype du script de mise à jour semi-automatique.

Lors de cette présentation, j'ai aussi fait un état des lieux des plugins proposés par GLPI pour rajouter une gestion des stocks, mais aucun n'est assez pertinent. J'ai donc proposé comme solution la création d'un plugin web en PHP, HTML* (HyperText Markup Language) et JS* (JavaScript) qui a été accepté. Ce plugin utilisera aussi une dépendance d'un autre plugin de nom « Generic Object ». Ce plugin sert à créer de nouveaux types d'équipements informatiques qui ne sont pas pris en charge directement par GLPI, tels que les claviers ou les souris, ou d'autres équipements informatiques qui ont besoin d'un espace dans GLPI à eux.

Dans cette 2ème et dernière partie du projet, il a fallu faire une phase de conception des interfaces que rajoutera ce plugin. Il y a donc 3 interfaces ajoutées par ce plugin. Ces maquettes* ont été faites grâce à l'application web Whismical*. Ces maquettes ont été créées de façon à respecter les besoins non fonctionnels du cahier des charges qui sont les critères de Bastien et Scapin*.

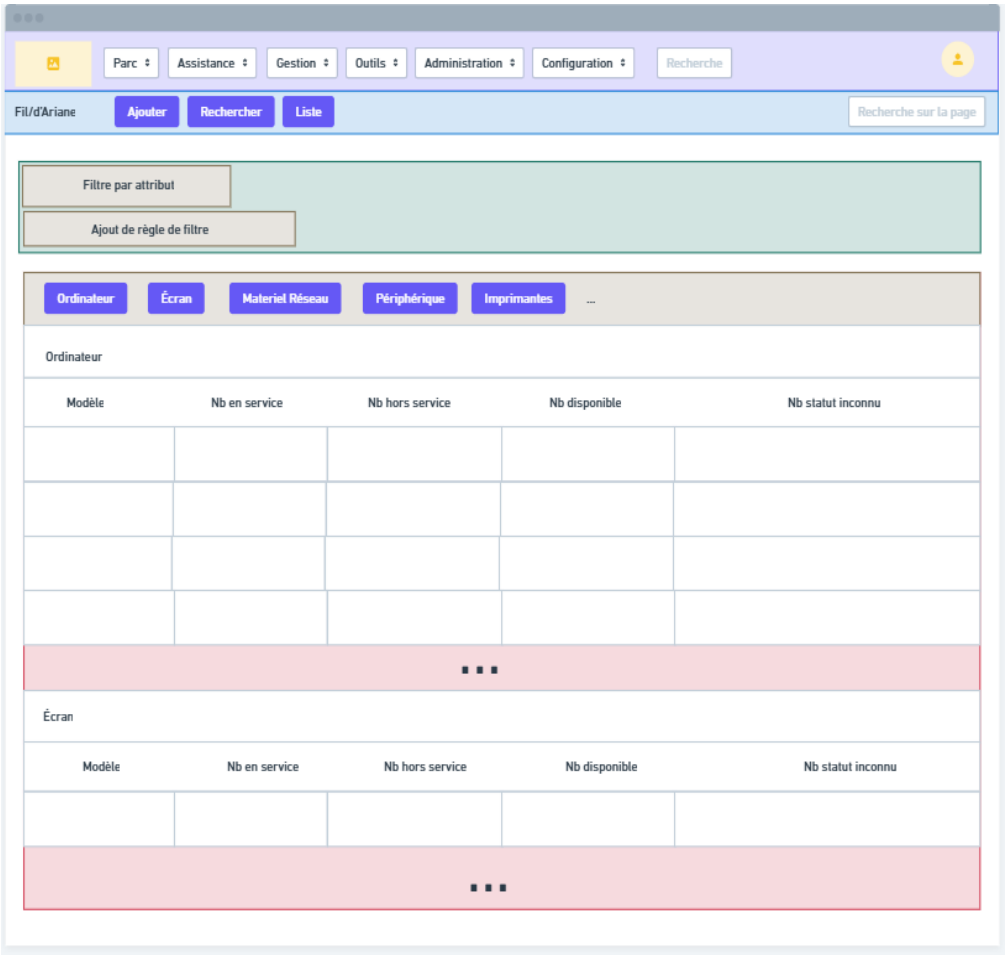


Figure 3 : Interface Stock

Pour la première interface, elle sert à afficher pour chaque type de matériel pris en charge par GLPI ou bien rajouté par le plugin « Generic Object ». J'affiche sur cette interface le modèle de l'équipement informatique ainsi que le nombre hors service, en service, disponible ou alors statut inconnu pour ce modèle, et cela pour chaque type d'équipement informatique. On retrouve tous les modèles d'ordinateur, d'écrans, de claviers, de souris, d'imprimantes et périphériques (clés USB, câbles, casques, etc). Cette page respecte les différents critères de Bastien et Scapin par son guidage à travers le fil d'ariane et son groupement des différents éléments, sa densité informationnelle avec uniquement les informations utiles et sa séparation entre les différents types d'équipements. Sur cette interface, on peut aussi filtrer les différents types d'équipement et rechercher les modèles que l'on veut. On peut aussi cliquer sur les différents modèles d'équipement qui nous envoient sur l'interface suivante.

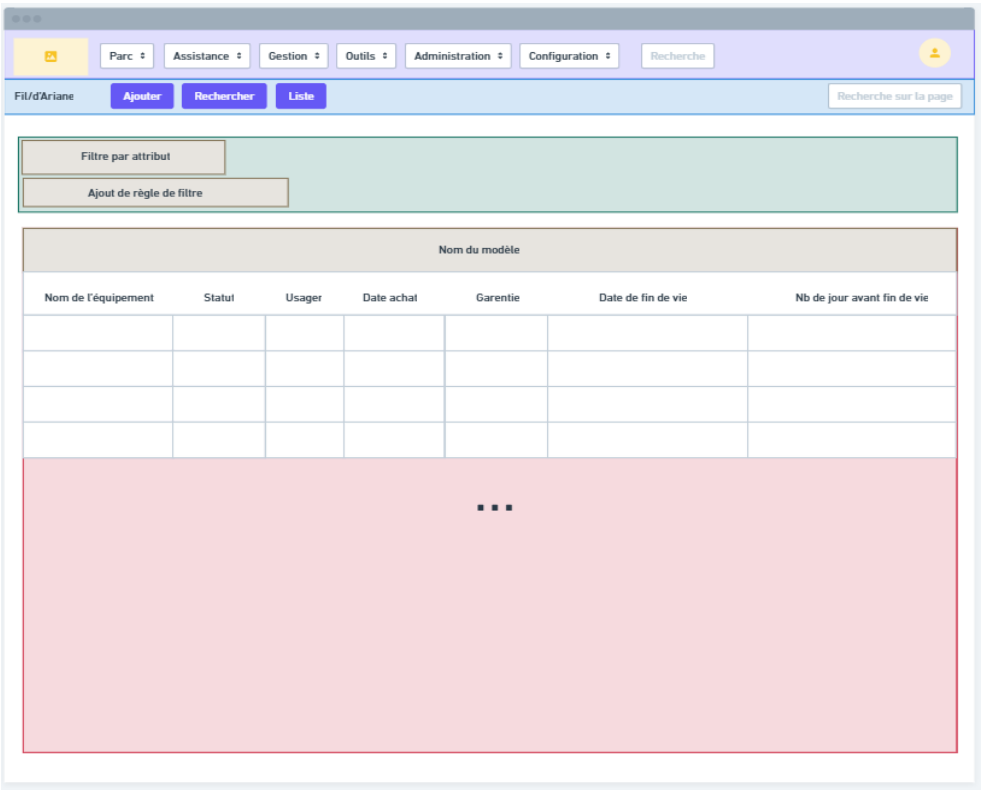


Figure 4 : Interface Stock par modèle

Sur cette 2ème interface qui se rapproche beaucoup de la première pour garder une certaine homogénéité demandée par le cahier des charges, mais cette fois-ci ce sera tous les équipements du modèle choisi avec leurs statuts, leurs cycles de vie ainsi que si son statut et non disponible car il est déjà prêté à quelqu'un, j'affiche le nom de la personne. On peut ensuite cliquer sur l'équipement qui nous envoie sur une page déjà définie par GLPI qui affiche ses caractéristiques.

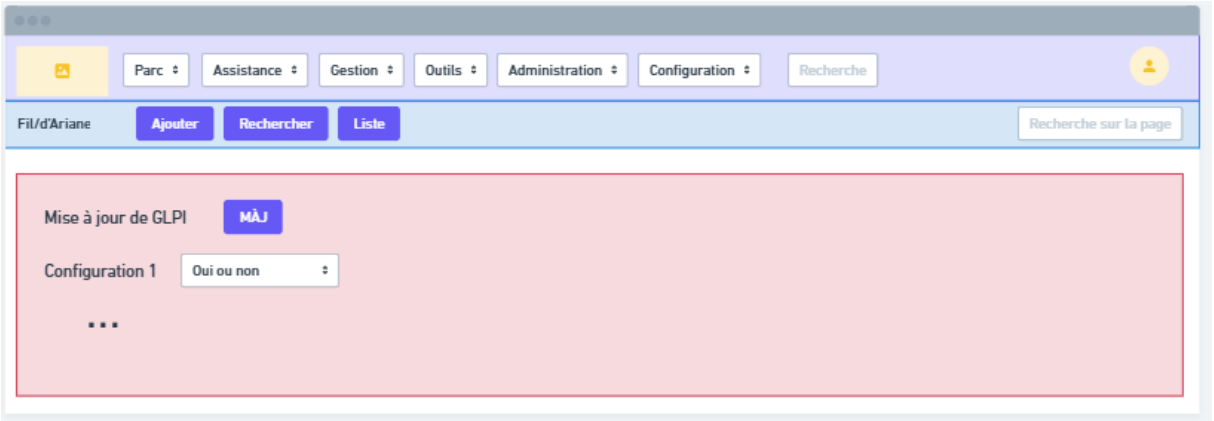


Figure 5 : Interface configuration du plugin

Sur cette 3ème interface, on retrouve la configuration du plugin, ses paramètres. On retrouve un bouton pour mettre à jour GLPI qui renvoie vers le Gitlab qui contient toutes les informations pour que celui qui veut mettre à jour GLPI puisse le faire.

IV.2 Réalisation

Côté technique, plusieurs choses sont à savoir. D'abord, en premier lieu, la structure d'un plugin GLPI qui est celle-ci ci-dessous.

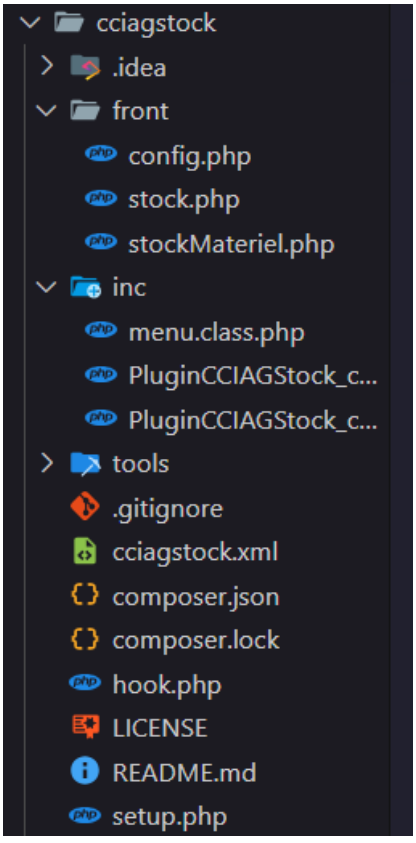


Figure 6 : Structure d'un projet plugin de GLPI

La structure d'un plugin GLPI ressemble particulièrement à une structure MVC*(Model View Controller) que l'on peut retrouver dans la création d'application Android ou de logiciel à quelque point près. Pour détailler la structure, nous avons la racine du projet qui est "cciagstock", c'est le nom de mon plugin GLPI. Le dossier ".idea" contient quelques fichiers de configuration pas très importants pour nous. Il y a ensuite le dossier "front" qui, en structure MVC, correspondrait au dossier "View" qui regroupera les interfaces web, "config.php" pour la page de configuration de mon plugin, "stock.php" c'est la page principale du plugin qui affiche, pour chaque modèle de chaque type d'appareil technologique (ordinateur, moniteur, souris, ...), cinq informations sont affichées qui sont les suivantes :

- Le nom du modèle
- Le nombre d'appareils de ce modèle en service
- Le nombre d'appareils de ce modèle hors service
- Le nombre d'appareil de ce modèle disponible en stock
- Le nombre d'appareils de ce modèle qui a un statut inconnu (valeur NULL dans la base de données)

Et enfin le 3ème fichier "stockMateriel.php" correspond à lorsque que l'on clique sur un des modèles depuis la page "stock.php" affiche tous les appareils de ce modèle avec des informations plus précises comme si l'ordinateur était en service, le nom de la personne à qui il a été prêt ou appartient ainsi qu'une fonctionnalité demandée par le client après ma présentation de mon projet, le cycle de vie de chaque appareil donc là on retrouvera la durée de la garantie, la date d'achat ainsi qu'une date de fin de vie qu'on aura au préalable enregistrée qui est de 4 ans par équipement informatique par défaut et enfin le temps qu'il reste avant de changer cet appareil.

On retrouve ensuite le dossier "inc" qui diffère un peu du modèle MVC car le dossier "inc" correspond au dossier "Modèle" et "Controller" dans une structure MVC donc dans ce dossier on retrouve le fichier "menu.class.php" ce fichier doit obligatoirement s'appeler comme ceci car GLPI ira chercher ce fichier automatiquement. Ce fichier correspond à l'ajout d'un menu dans un autre menu de GLPI. Ici, dans mon plugin, j'ajoute un menu "Stock" au menu "Tools" de GLPI le deuxième fichier "PluginCCIAGStock_config.php" contient les fonctions qui sont exécutées depuis la page "config.php" ainsi que la fonction "install()" qui est très importante. Cette fonction dans mon plugin sert à créer des "vues" dans la base de données qui contiennent les données que j'affiche sur la page "stock.php" et enfin le 3ème fichier "PluginCCIAGStock_controller" qui correspond au contrôleur principal de la page "stock.php" et "stockMateriel.php". On retrouve dans ce fichier une fonction getAllMaterial() qui va chercher dans la base de données les données qui ont besoin d'être affichées sur les deux interfaces.

On retrouve ensuite 2 fichiers importants : le « setup.php » et le « hook.php ». Ces 2 fichiers sont essentiels pour l'installation et le lancement du plugin GLPI. Le fichier « setup.php » contient la plupart des fonctions utiles et obligatoires pour mon plugin que GLPI détecte automatiquement s'ils sont bien nommés. La façon standard de les nommer est « plugin_NOM-DU-PLUGIN_NOM-DE-LA-FONCTION() » sachant que « NOM-DE-LA-FONCTION » doit avoir comme nom une fonction reconnue par GLPI pour qu'il puisse l'exécuter. Le fichier « hook.php » contient des fonctions utiles pour l'ensemble de mon plugin.

Deuxièmement, créer ce plugin demande un travail de compréhension pour comprendre et utiliser l'API de GLPI qui a été compliquée à prendre en main par sa multitude de fonctions, d'objets et la façon dont GLPI appelle les plugins. Lorsque l'on va dans l'onglet plugins de GLPI, on peut installer un plugin. Lorsqu'on installe un plugin, GLPI va exécuter plusieurs fonctions dans le code de GLPI, ainsi qu'une fonction importante pour nous situer dans le "hook.php" du plugin qui est cette fonction :

```
static function install() {
//static function install(Migration $mig) {

    global $DB, $CFG_GLPI;
    $types = $CFG_GLPI["asset_types"];
    foreach ($types as $type) {
        $table = strtolower($type);
        if ($table != 'softwarelicense' && $table != 'certificate' && $table != 'unmanaged' && $table != 'appliance') {
            if ($DB->tableExists("glpi_{$table}s")) {
            $sql = "CREATE OR REPLACE VIEW `vue_stock_{$table}` AS
                    SELECT m.name AS model,
                           SUM(CASE WHEN s.name = 'En Service' THEN 1 ELSE 0 END) AS ES,
                           SUM(CASE WHEN s.name = 'Hors Service' THEN 1 ELSE 0 END) AS HS,
                           SUM(CASE WHEN s.name = 'Stock SI' THEN 1 ELSE 0 END) AS Disponible,
                           SUM(CASE WHEN s.name IS NULL THEN 1 ELSE 0 END) AS statut_inconnu
                    FROM `glpi_{$table}s` mat
                    LEFT JOIN `glpi_{$table}models` m ON mat.`{$table}models_id` = m.id
                    LEFT JOIN `glpi_states` s ON mat.states_id = s.id
                    GROUP BY m.name";
            $DB->queryOrDie($sql, "Erreur lors de la création de la vue [{$table}]."."<br>".$DB->error());
            }
            // else {
            // if (!$DB->fieldExists($table, "VOTRE_ATTRIBUT_A_RAJOUTER")
            // $mig->addField($table, 'VOTRE_ATTRIBUT_A_RAJOUTER', 'TYPE_ATTRIBUT', ['value' => VALEUR_PAR_DEFAUT (pas oblig
        }
    }
    // $mig->executeMigration();
    return true;
}
```

Figure 7 : Fonction « install() » de mon plugin CCIAGStock

Elle s'exécutera uniquement lors de l'installation. Cette fonction a pour but de modifier la base de données en rajoutant des vues* qui nous serviront pour les différentes interfaces.

Ce qu'il faut savoir, c'est que cette fonction utilise une variable de GLPI nommée « CFG_GLPI » avec l'attribut « asset_type » appartenant à cette variable. Les valeurs se trouvant à cet attribut sont tous les types d'équipements informatiques gérés par GLPI, donc les ordinateurs, écrans, logiciels et plein d'autres. Donc pour chacun de ces types, je vais leur créer une vue uniquement si ces équipements peuvent avoir un modèle, comme « Samsung G5 Odyssey » qui est un modèle d'écran de l'entreprise Samsung. Ce qui n'est pas le cas pour les licences de logiciel, les certificats, les objets non gérés par GLPI automatiquement et les appareils en général. Pour les autres équipements, je leur crée donc une vue dans la base de données à laquelle ces vues auront des attributs correspondant aux données qui sont celles expliquées plus haut dans le mémoire de stage. La partie commentée quant à elle permet de rajouter des attributs si l'utilisateur du plugin souhaite rajouter de nouvelles données.

Après l'installation le plugin est désactivé par défaut, on l'active et au moment où on active le plugin, GLPI va exécuter automatiquement des fonctions qui sont appelées dans un certain ordre. La première et la plus importante, c'est celle-ci :

```php
function plugin_init_cciagstock()
{
    global $PLUGIN_HOOKS;

    error_log("DEBUG: Path to hook.php is: " . __DIR__ . '/hook.php');
    include_once(__DIR__ . '/hook.php');
    $PLUGIN_HOOKS['menu_toadd']['cciagstock'] = 'plugin_cciagstock_menu_toadd';

    Plugin::registerClass('PluginCCIAGStockMenu', ['filepath' => 'inc/menu.class.php']);
    $PLUGIN_HOOKS['csrf_compliant']['cciagstock'] = true;
    $PLUGIN_HOOKS['config_page']['cciagstock'] = 'front/config.php';
}
```

Figure 8 : Fonction « plugin_init_cciagstock() » de mon plugin CCIAGStock

Cette fonction est la première à s'exécuter et elle s'exécutera à chaque changement d'interface ou de rechargement de page web. Cette fonction contient ce que j'appelle les prérequis du plugin pour son bon fonctionnement, on retrouve actuellement pour mon plugin :

- La variable $PLUGIN_HOOKS, comme son nom l'indique, est un crochet qui va attraper certaines valeurs et les attribuer aux différents attributs de GLPI. Cette variable provient directement de GLPI.
- La ligne « include_once(__DIR__.'/hook,php') correspond à l'inclusion du fichier hook.php dans setup.php, ce qui me permet d'utiliser mes fonctions situées dans ce fichier.
- La « ligne $PLUGIN_HOOKS['menu_toadd']['cciagstock'] = 'plugin_cciagstock_menu_toadd'; » correspond au fait que je récupère le résultat de ma fonction plugin_cciagstock_menu_toadd dont le résultat est une variable menu avec son nom et le lien de la page visée et je le rajoute à l'attribut ['menu_toadd'] de PLUGIN_HOOKS qui s'occupe des menus situés dans la barre de navigation de GLPI.
- La prochaine ligne correspond à l'affectation de ma classe PluginCCIAGStockMenu dans l'objet Plugin de GLPI pour que ma classe soit reconnue et utilisable par GLPI et je lui indique où mon fichier contenant la classe se trouve
- • La ligne « $PLUGIN_HOOK['csrf_compliant']['cciagstock'] = true ; » correspond au fait que mon plugin utilise le CSRF* (Cross-Site Request Forgery) de GLPI pour protéger mon plugin.
- La dernière ligne correspond à l'ajout d'une page de configuration pour mon plugin.

En conclusion de cette partie, au moment où j'écris ce mémoire de stage, cette partie n'est pas encore finie et implémentée. Il peut y avoir plus de fonctions ou de mécaniques GLPI qui peuvent faire que mon code ne marche plus et que les images de code utilisées pour commenter le fonctionnement de GLPI ne fonctionnent pas. Donc les fonctions fonctionnent, mais peut-être il est possible que je les change plus tard.

V. CONCLUSION

Au cours de ce stage au sein de la Compagnie de Chauffage Intercommunal de l'Agglomération de Grenoble, j'ai pu découvrir le travail en entreprise au sein d'une équipe informatique et découvrir leur quotidien. Ce stage m'a aussi permis de mettre en avant les compétences acquises par la formation de BUT INFO depuis 2 ans. La réalisation de cette mission m'a aussi permis d'étendre mes compétences dans beaucoup de domaines que ce soit en réseau, en développement web, en base de données ou encore en gestion de projet et de communication en entreprise. J'ai trouvé cette expérience en entreprise très intéressante et enrichissante, ce qui me permet de finir mon stage sur une note particulièrement positive.

VI. RÉSUMÉ COURT EN FRANÇAIS ET ABSTRACT EN ANGLAIS

VI.1 Abstract

Why can adding a life cycle for hardware equipment improve the logistics of CCIAG digital services

Jarod Tivollier

In 2025, the digital service uses a web application name by GLPI for ticket management and inventory for certain computer hardware such as computers, monitors, network equipment, or printers. Other materials like mouse and keyboard are written in an excel file, but whether you are on GLPI or on the excel file, nothing says when the hardware equipment is becoming too old. To resolve this problem, we make a life cycle for hardware from purchase date to number of years of service we gave to this equipment. We make two new objects in GLPI, one for the mouse and the other for the keyboard, to centralize all equipment in one place after we get purchase date and guarantee on this equipment, we choose a date that will allow to say that this object has made its life cycle and it is time to replace it all of this is made by a GLPI plugin we made. Creating this helps the digital service to make fewer repetitive, and longer tasks, allows for more in-depth monitoring of equipment and to prevent problems related to the age of the equipment.

Keyword: management, plugin, GLPI, hardware, life cycle, IT

VI.2 Résumé court en français

Au cours de ce stage au sein de l'entreprise CCIAG dans leur service numérique dans le cadre de la 2ème année du BUT informatique. J'ai eu pour mission « Évolution et Amélioration de GLPI » qui consistait à mettre à jour GLPI et rajouter une gestion des stock plus complète en rajoutant les équipement informatique non pris en charge par GLPI et leur ajouter un cycle de vie dans un premier temps après l'élaboration du cahier des charges j'ai crée une VM qui accueillera GLPI et sa base de données auparavant GLPI et sa base de données était héberger sur Windows j'ai décidé d'opter pour un hébergement sous Linux avec l'aide du logiciel Docker pour mettre à jour GLPI j'ai décidé de crée un script de mise à jour semi-automatique de GLPI et de sa base de données en Bash qui permet la persistance des données ainsi que l'ajout de potentiel plugin. Cette partie permet donc de faire la mise à jour de GLPI puis vient ensuite l'amélioration de la gestion de stock pour cela j'ai crée un plugin pour GLPI en PHP, JS et HTML qui vient rajouter quelque interface pour afficher cette gestion des stocks ainsi que le cycle de vie pour chaque équipement informatique contenu dans la base de données de GLPI.

VII.GLOSSAIRE

API (Application Programming Interface) : C’est un ensemble de définitions et de protocoles qui permet à différentes applications logicielles de communiquer entre elles.

Apache : C’est un serveur web open source permettant de charger des pages web et d'autres contenus (images, vidéos, etc.) dans un navigateur.

Bash : C’est un langage de script.

Cache : est une couche de stockage de données grande vitesse qui stocke un sous-ensemble de données.

CCIAG (Compagnie de Chauffage Intercommunale de l’Agglomération de Grenoble) : C’est une entreprise spécialisée dans le chauffage urbain ou les réseaux de chaleur de l'agglomération de Grenoble.

Clé d’accès : C’est un code alphanumérique ou une chaîne de caractères utilisée pour authentifier et autoriser l'accès à un système, une ressource, une application ou un service.

Conteneur : C’est une unité logicielle standardisée et légère qui regroupe toutes les dépendances nécessaires à l'exécution d'une application : le code, le runtime, les bibliothèques système, les outils, et les configurations.

Critères de Bastien et Scapin : C’est un ensemble de règles et de principes utilisés pour évaluer la qualité ergonomique d'une interface utilisateur.

CPU (Central Process Unit) : Processeur d’un ordinateur.

CSRF (Cross-Site Request Forgery) : C’est la falsification de demande intersite (CSRF ou XSRF) fait référence à une attaque qui fait que l’utilisateur final effectue des actions indésirables dans une application Web qui lui a déjà accordé l’authentification.

Docker : Logiciel permettant à un environnement informatique d’héberger des conteneurs et des volumes.

Dockerfile : C’est un fichier exécuté par Docker contenant des instructions pour la création des conteneurs.

Équivalent logements : C’est une unité de mesure utilisée, notamment dans le domaine du chauffage urbain ou des réseaux de chaleur, pour quantifier la puissance thermique ou la consommation d'énergie nécessaire pour chauffer un ensemble de bâtiments, en la rapportant à un nombre standard de logements résidentiels.

GitHub : C’est une plateforme web très populaire qui fournit un service d'hébergement de dépôts de code source utilisant le système de contrôle de version Git.

GLPI (Gestionnaire Libre de Parc Informatique) : C’est une application web permettant de gérer son inventaire d’équipement informatique et de créer une gestion de ticket pour pouvoir créer des tickets lorsqu’on a un problème.

GPU (Graphic Process Unit) : Carte graphique d'un ordinateur.

HTML (HyperText Markup Language) : C'est le langage de balisage standard utilisé pour créer et structurer le contenu des pages web.

JS (JavaScript) : C'est un langage de programmation principalement utilisé pour rendre les pages web interactives et dynamiques côté client (dans le navigateur de l'utilisateur).

Maquette : C'est une représentation visuelle statique et non fonctionnelle d'une interface utilisateur (pour un site web, une application mobile, un logiciel, etc.).

Marketplace : Nom donné à une interface où se regroupent des créations d'utilisateurs.

MVC (Model, View, Controller) : est un modèle dans la conception de logiciels. Il met l'accent sur la séparation entre la logique métier et l'affichage du logiciel.

PHP (Hypertext Preprocessor) : C'est un langage de script côté serveur principalement utilisé pour le développement web.

Plugin : C'est un composant logiciel qui ajoute des fonctionnalités spécifiques à un programme informatique existant, sans modifier le code source de ce programme principal.

PostgreSQL : C'est un système de gestion de base de données relationnelle (SGBDR) open source.

RAM (Random Access Memory) : Mémoire d'un ordinateur.

Regex : C'est une séquence de caractères qui définit un modèle de recherche.

SEML (Société d'Économie Mixte Locale) : C'est une entreprise dont le capital est détenu conjointement par des collectivités territoriales (communes, départements, régions) et des partenaires privés.

Service : Fait référence à un programme ou une application qui s'exécute en arrière-plan sur un système d'exploitation.

Soluciel : mot employé par le service numérique de la CCIAG pour dire Solution Logiciel

Token : C'est une séquence de caractères numériques ou alphanumériques qui représente quelque chose ; dans votre projet, cela correspond à un droit d'authentification et d'utilisation de l'API GitHub.

URL (Uniform Resource Locator) : C'est une chaîne de caractères qui identifie de manière unique l'emplacement d'une ressource sur Internet (comme une page web, une image, une vidéo, un fichier) et la manière d'y accéder.

VM (Virtual Machine) : C'est un environnement informatique virtuel qui émule un système informatique complet (ordinateur).

VMware vSphere : C’est une suite logicielle de virtualisation de serveurs développée par VMware. C'est une plateforme de virtualisation de pointe qui permet aux entreprises de transformer leurs ressources informatiques physiques en une infrastructure virtuelle.

Volume : C’est un bloc de données que l’on peut relier à des conteneurs sans que les données soient affectées par la suppression du conteneur.

Vue : C’est un objet de base de données comme un table qui est définie par une requête SQL qui agrège, filtre ou combine des données provenant de plusieurs tables.

Whimsical : C’est une application web permettant la création de maquettes d’interface.

VIII. ANNEXES

VIII.1Annexe A : Références bibliographiques et webographiques

[1] Article d'actualité du chauffage urbain de Grenoble sur la chaufferie Biomax – Disponible sur : https://www.chauffage-urbain-grenoble.fr/3820-biomax-tout-savoir-sur-le-nouveau-site-de-production.htm (Consulté le 23 mai 2025)

[2] Article d'actualité de la CCIAG sur l'UIVE Athanor – Disponible sur : L'Usine d'Incinération et de Valorisation Énergétique Athanor - Compagnie de chauffage (Consulté le 3 juin 2025)

[3] Site de la CCIAG sur la page Histoire de l'entreprise - Disponible sur : Histoire - Compagnie de chauffage (Consulté le 3 juin 2025)

[4] Article d'actualité du chauffage urbain de Grenoble sur la chaufferie de Villeneuve – Disponible sur : Notre site de production Villeneuve - Chauffage Urbain (Consulté le 3 juin 2025)

[5] Page Wikipedia de la CCIAG – Disponible sur : Compagnie de chauffage intercommunale de l'agglomération grenobloise — Wikipédia (Consulté le 2 mai 2025)

[6] Correcteur orthographique du site Reverso – Disponible sur Correcteur d'orthographe et de grammaire - Français - Reverso (Consulté le 4 juin 2025)

VIII.2Annexe B : Cahier des charges

Cahier des charges

Projet de mise à niveau et évolution de GLPI

« Projet GLPI 2030 »

Présenté par

Jarod Tivollier

Sommaire

I. Présentation du projet

Le service numérique de la Compagnie de Chauffage Intercommunale de l'Agglomération de Grenoble (CCIAG) utilise des ordinateurs sous Windows 10 ou Windows 11 avec plusieurs logiciels installés, dont l'application web GLPI (Gestionnaire Libre de Parc Informatique) qui permet de gérer différentes tâches telles que la gestion de tickets. Ils utilisent cette application web quotidiennement et ils y sont habitués. On retrouve dans leur GLPI plusieurs tableaux de bord contenant les mêmes informations mais agencé différemment

GLPI est une application web libre de gestion de services informatiques créé par l'association INDEPNET, qui rendit le projet open source pour qu'une communauté de développeurs puisse venir modifier ou développer le code ainsi que créer des plugins permettant d'ajouter des fonctionnalités spécifiques à certains besoins. Cette application web utilise principalement comme langage de programmation PHP et JavaScript ainsi que du langage de balisage HTML et XML. L'application web peut aussi utiliser une base de données qui doit être soit en MySQL ou bien MariaDB.

Actuellement le service numérique utilise la version 10.0.2 de GLPI, alors que la dernière version de GLPI est 10.0.18, soit 16 versions de retard ce qui correspond à environ 3 ans de mise à jour. Les plugins utilisés sont :

- **Champs supplémentaires v1.21.6** : plugin permettant l'ajout de champs en plus pour diverses fonctionnalités telles que les tickets.
- **Comportements v 2.7.1** : plugin permettant l'ajout de comportements à GLPI.
- **Imports fabricants v3.0.5** : plugin permettant l'import de données financières comme la date d'achat, la garantie ou encore une page HTML des fabricants d'un ordinateur (fonctionne avec Dell, HP, Toshiba, Lenovo (<= 1.5.0), Fujitsu-Siemens et Wortmann AG).
- **Oauth IMAP v1.4.2** : plugin permettant d'avoir une authentification utilisateur par IMAP (Protocole d'accès à la messagerie Internet).
- **Plus de rapports v1.8.5** : plugin permettant d'ajouter différents tableaux de statistiques.
- **Tableau de bord v1.0.2** : plugin permettant d'avoir un tableau de bord contenant diverses informations.

A) Analyse des problèmes actuels et hypothèses des solutions

Actuellement, si on veut passer sur la dernière version de GLPI, c'est possible, mais là où cela va coincer, c'est au niveau des plugins. Le plugin « Tableau de bord » créé par Stevenes Donato n'est plus maintenu à jour et est même supprimé, on ne peut plus l'installer. Pour remédier à cela, il faudra trouver une alternative à ce plugin en le remplaçant par un autre **voire le supprimer car GLPI possède déjà un autre tableau de bord.**

Un autre problème est la version de GLPI actuellement. La version de GLPI utilisée est la 10.0.2 qui fonctionne très bien, mais cette version date de 2022. Le service numérique possède donc une dette technologique sur leur logiciel. En montant la version, premièrement les possibles failles de sécurité du logiciel ont été résolues. Deuxièmement ils pourront bénéficier des dernières fonctionnalités de GLPI qui peuvent être intéressantes et utiles au service numérique.

B) Objectif

Ce projet a pour objectif de mettre à niveau le GLPI dans sa version la plus récente ainsi que ses plugins s'ils sont disponibles sur la version la plus récente et compatibles avec. Déterminer et remplacer l'existant par un plan de migration, Cela engendre de passer par une migration de données, d'ajouter des fonctionnalités telles que des nouveaux plugins en particulier un plugin de gestion des stocks qui est le besoin principal avec la montée de version de GLPI, des modifications des interfaces pour s'adapter aux critères de Bastien et Scapin (ensemble de critères ergonomiques pour l'évaluation d'interfaces utilisateur) et enfin vérifier le bon fonctionnement après toutes les modifications.

Pour résumé cela voici le **QQOQCCP** correspondant au projet :

Qui ? : Ce projet vise à répondre à un besoin du service numérique de la CCIAG.

Quoi ? : Ce projet vise à mettre à jour le GLPI ainsi que l'améliorer.

Où ? : Ce projet est produit au 25 Av. de Constantine à Grenoble au Polynôme.

Quand ? : Ce projet commence le 15 avril 2025 et ce finit au plus tard le 20 juin 2025.

Comment ? : Ce projet utilisera une VM crée depuis VMWare Sphere où il hébergera la version la plus récente de GLPI pour permettre d'apporter des modifications et des ajouts en toute sécurité et intégrité. Puis de vérifier par des tests fonctionnels et utilisateur que tout fonctionne parfaitement et que cette nouvelle version du GLPI est adapté au client.

Combien ? : Ce projet ne demande aucun financement.

Pourquoi ? : Ce projet a pour cause un besoin principal d'avoir une gestion des stocks sur GLPI mais aussi une dette technologique du service numérique sur cette application web ainsi qu'un besoin d'amélioration en rajoutant des fonctionnalités utiles et rendant une tâche plus facile.

II. Besoins fonctionnels

A) Besoins fonctionnels

Les besoins fonctionnels requis pour ce projet sont ceux qui sont déjà existants, donc par exemple les tickets où on va pouvoir créer des tickets, supprimer des tickets, leur ajouter une description et plein d'autres informations. Ce projet a pour but principal dans un premier temps de rajouter un plugin de gestion de stock pour que le client (SN) puisse gérer son stock d'ordinateurs, d'écrans, câbles, etc. Les cas d'utilisation suivants représentent ce que peuvent faire les différents acteurs actuellement. Ils sont représentés dans un diagramme de cas d'utilisation en annexe du cahier des charges.

Acteur : Technician

Package « Stock » :

- Visualiser la Gestion des stocks **UC1**
- Visualiser le nombre et informations des appareils restant en stock **UC2**
- Visualiser le nombre et informations des appareils en activité **UC3**
- Visualiser le nombre et informations des appareils HS **UC4**
- Visualiser le nombre et informations des appareils par lieu **UC5**
- Filtrer les informations des appareils **UC6**

Sur le GLPI sur service numérique il existe 4 rôles dont le rôle super-admin et admin qui ne sont pas représentés dans le diagramme car ils ont tous les droits.

Ces cas d'utilisation ne sont qu'une représentation théorique des différentes fonctionnalités qui peuvent être ajoutées en fonction de comment avance le projet. Il est possible que des fonctionnalités s'ajoutent ou bien s'enlèvent.

B) Besoins non-fonctionnels

Pour satisfaire et aider le client dans l'utilisation du logiciel il faut pour cela respecter certain critère ergonomique qui sont les suivants :

Guidage :

- **Groupement des éléments** :
  - Il est important que les informations afficher sur l'interface correspondent entre elles il faut donc les grouper en fonction de l'information qu'elle renvoie je ne peux pas mettre le nombre d'ordinateur actif avec la température actuel des différents ordinateurs.
- **Lisibilité** :
  - Il est important que chaque information soit visible et espacée pour permettre une compréhension plus facile.

Charge de travail :

- **Densité informationnelle** :
    - Il est important qu'une interface ne contienne pas trop d'information cela pourrait nuire à la compréhension de l'utilisateur.

Homogénéité :

- Il est important de maintenir une certaine cohérence sur les différentes interfaces pour ne pas perdre l'utilisateur.

Gestion des erreurs :

- Dans notre cas il est important que s'il y a un problème avec la base de données que l'utilisateur sois prévenu qu'il ne peut pas consultée les informations actuellement.

Adaptabilité :

- **Prise en compte de l'expérience utilisateur :**
    - Une interface doit pourvoir répondre à l'expérience personnalisé de l'utilisateur le plus possible en lui permettant d'afficher ou de déplacer des informations comme il le souhaite pour pouvoir adapter l'interface selon ses besoins.

Ces critères ergonomiques correspondent aux critères de Bastien et Scapin qui ont pour but de déterminer l'utilisabilité d'une interface.

III. Contraintes et risques

A) Tableau des contraintes et des risques

Le tableau suivant correspond aux différentes contraintes liées au projet ainsi que les risques que ces contraintes peuvent avoir.

Contraintes		**Risques induits**			
Types de contrainte	Nom de la contrainte	Non et index du risque	Type de risque	Criticité (impact * probabilité)	Mitigation
Temporelles	Dates des livrables	Retard sur le projet **[TP01]**	Temporel	2*2	*Protection* Poser des dates limites pour chaque tâche et surveiller leur avancée
Humaine	Equipe	Malade **[H01]**	Humain	2*1	*Accepter* Accepter le fait de rater des heures de travail (valable surtout sur les projets « solo ») et de revoir son emploi du temps *Protection* Prévoir un plan de secours et répartir les tâches si besoin
Organisationnelle	Demande du client	S'éloigner de la demande du client en cas de mauvais cadrage du projet **[O01]**	Organisationnelle	3*2	*Reserve* Faire des points réguliers avec les clients pour pouvoir recadrer le projet en cas de dérive
	Equipe	RTT ou CP **[O02]**	Organisationnel	1*3	*Protection* Poser les questions avant les jours de RRT ou CP

					Recherche Rechercher comment résoudre le problème *Partage* Demander de l'aide à d'autres personnes du service
Technique	GLPI	Manque de droits pour accéder à certaines informations **[TE01]**	Technique	1*3	*Partage* Demander les droits aux personnes pouvant les donner

Certains risques ne découlent pas directement de contraintes telles que les suivantes :

Risques			
Nom et index du risque	Type de risque	Criticité (Impact * Probabilité)	Mitigation
Maîtrise lacunaire sur certaines tâches **[TE02]**	Technique	2*2	*Recherche* Lire des documentations *Partage* Demander de l'aide dans le cas où je ne trouve pas de solutions
Problème sur la VM **[TE03]**	Technique	3*1	*Recherche* Rechercher une possible solution au problème *Accepter* Problème important sur la VM nécessitant sa destruction et reconstruction

B) Matrice d'incidence aux risques

Incidence	Impact		
Probabilité		H01	TE03
		T01, TE02	O01
	O02, TE01		

IV. Organisation de l’équipe projet

Pour que le projet se déroule de la façon la plus efficace possible, il faut répartir les tâches en fonction des compétences de chacun. Voici une présentation des membres de l’équipe pour ce projet :

Jarod Tivollier :

- **Compétences** : compétence en base de données, Réseau et dev web
- **Rôles** : chef de projet, suivre l’avancée du projet et maitre d’œuvre

Victor Traumann :

- **Rôles** : Directeur de projet

Thomas Nicolas :

- **Compétences** : compétence en base de données
- **Rôles** : Référent bases données

Romain Faccio :

- **Compétences** : compétence en base de données
- **Rôles** : Référent bases données

Nicolas Gonnon :

- **Compétences** : compétence en réseau
- **Rôles** : Référent infrastructure et systèmes

Laurent Tollar :

- **Compétences** : compétence en réseau
- **Rôles** : Référent infrastructure et systèmes

Cyril Brun

- **Rôles** : Référent outils

V. Plan d'action

Étape 1 : Organisation et prise de connaissance :

Dans un premier temps, il nous faut prendre connaissance du projet et créer un cahier des charges.

Étape 2 : Préparation de l'environnement de travail :

Dans un deuxième temps, après avoir pris connaissance du projet, il nous faut avoir un environnement adéquat pour le GLPI. Pour cela, nous allons créer une VM avec VMWare Sphere. Il nous faudra réfléchir à quelle OS est le plus optimisé pour le déploiement et le maintien de GLPI et enfin installer GLPI et le configurer.

Étape 3 : Accès à la base de données depuis la VM

Dans un troisième temps, après avoir un environnement de travail fonctionnel et avec GLPI dessus, il nous faudra pouvoir accéder à la base de données et tester que tout fonctionne.

Étape 4 : Recherche de potentiels plugins

Dans un quatrième temps, nous allons effectuer des recherches sur des plugins qui peuvent être intéressants et utiles pour le client dont en priorité un qui permet de faire une gestion des stocks.

Étape 5 : Ajout des plugins, modification de l'existant et ajout de fonctionnalités

Dans un cinquième temps, cette étape risque d'être la plus longue. Elle comprend l'ajout des plugins trouvés, la modification possible des interfaces ainsi que du schéma de base de données (mettre en étoile si ce n'est pas déjà fait et donc changement des ETL) et l'ajout de fonctionnalité si pertinent. Quand toute cette partie sera finie, il faudra vérifier que tout fonctionne par des tests fonctionnels et donc de sortir un prototype de l'application.

Étape 6 : Tests utilisateurs

Dans un sixième temps, quand tout fonctionnera, il nous faudra faire des tests utilisateur pour savoir si cela respecte bien les conditions du cahier des charges en termes d'interfaces et juger si les différentes fonctionnalités ajoutées leur semblent utiles.

Étape 7 : Modification ou/et correction

Dans un dernier temps il nous faudra modifier et/ou corriger les problèmes qu'auront trouver les utilisateurs qui ont testé notre application puis répéter l'étape 6 et 7 jusqu'à que le client valide le prototype du nouveau GLPI.

Étape Bonus : Création d'un plugin

Cette étape est bonus, elle sera réalisée oui ou non en fonction de l’avancement des autres étapes, mais cette étape aurait pour but de créer un plugin pour empêcher certaines interfaces changées manuellement d’être écrasées lors des mises à pour de GLPI.

VI. Planification

A) Organisation

Pour la planification du projet, j'utiliserais l'application web Jira qui me servira à créer et maintenir le diagramme de Gantt (diagramme représentant les différentes tâches avec leurs durées) ainsi qu'un possible tableau de Kanban me permettant de savoir où en est chaque tâche (À faire, En cours, À tester, Finis) vous trouverez une image du diagramme de Gantt en annexe.

B) Suivi du projet

Pour le suivi du projet, cela se fera par des petites présentations de 5-10 min lors des « Daily » à 9h30 pour présenter l'avancement du projet. Il pourra y avoir des présentations avec support ou bien sans quand cela est nécessaire.

VII. Sitographie

[1] Wikipédia de GLPI

Gestionnaire Libre de Parc Informatique — Wikipédia

[2] Le site officiel de GLPI

Accueil - GLPI Project

[3] Github du plugin « Champs supplémentaires »

GitHub - pluginsGLPI/fields: Additionals fields for GLPI

[4] Github du plugin « Comportements »

GitHub - yllen/behaviors: This plugin allows you to add optional behaviors to GLPI.

[5] Github du plugin « Imports fabricants »

GitHub - InfotelGLPI/manufacturersimports: Plugin manufacturersimports for GLPI

[6] Github du plugin « Oauth IMAP »

GitHub - pluginsGLPI/oauthimap: Oauth authentication for Imap receivers of GLPI

[7] Github du plugin « Plus de rapports »

GitHub - pluginsGLPI/mreporting: Mreporting plugin (Download : https://github.com/pluginsGLPI/mreporting/releases)

[8] Page du plugin « Tableau de bord » (Github plus disponible)

GLPi Plugins

[9] Jira (diagramme de Gantt)

Jira | Logiciel de suivi des tickets et des projets | Atlassian

[10] Draw.io (diagramme cas d'utilisation)

Diagramme sans nom - draw.io

[11] Reverso (Correcteur orthographique et de grammaire)

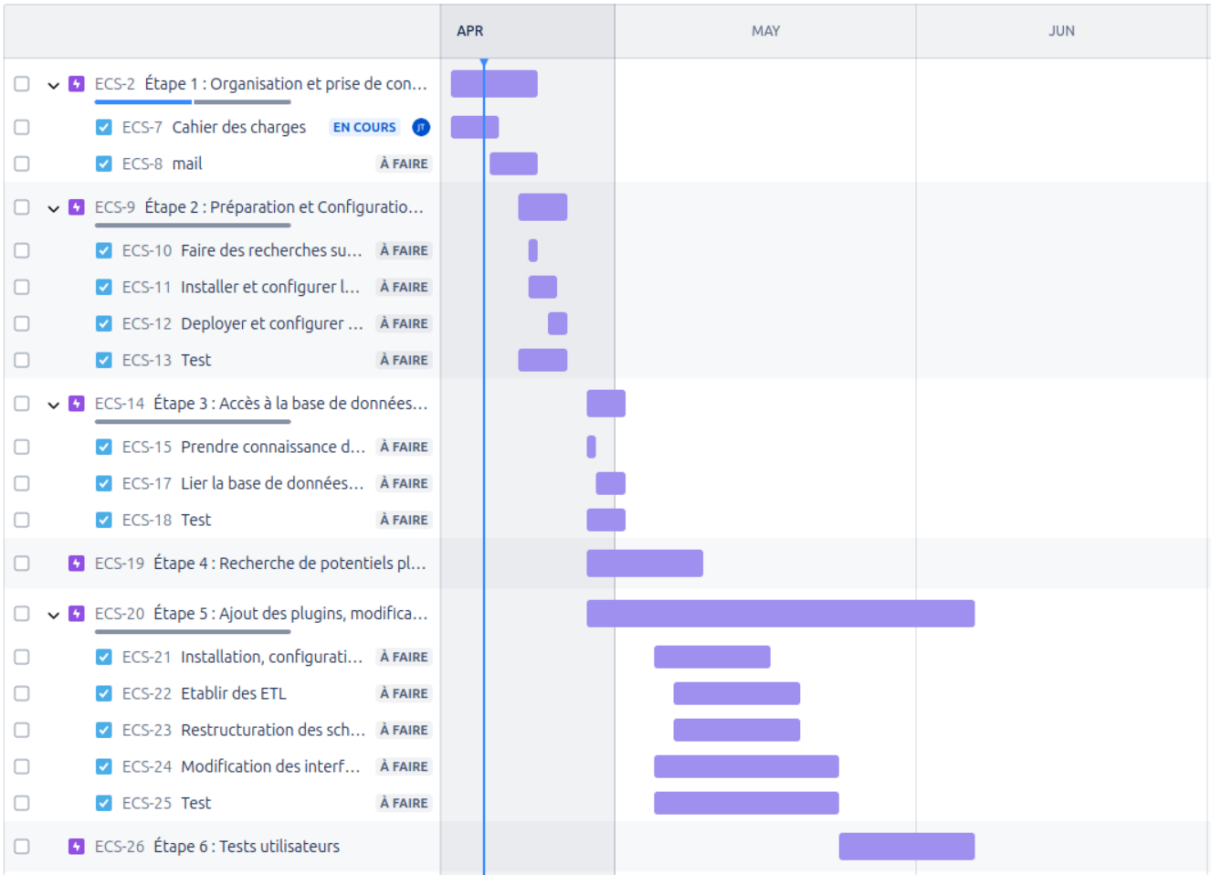
Correcteur d'orthographe et de grammaire - Français - Reverso

[12] GLPI du service numérique

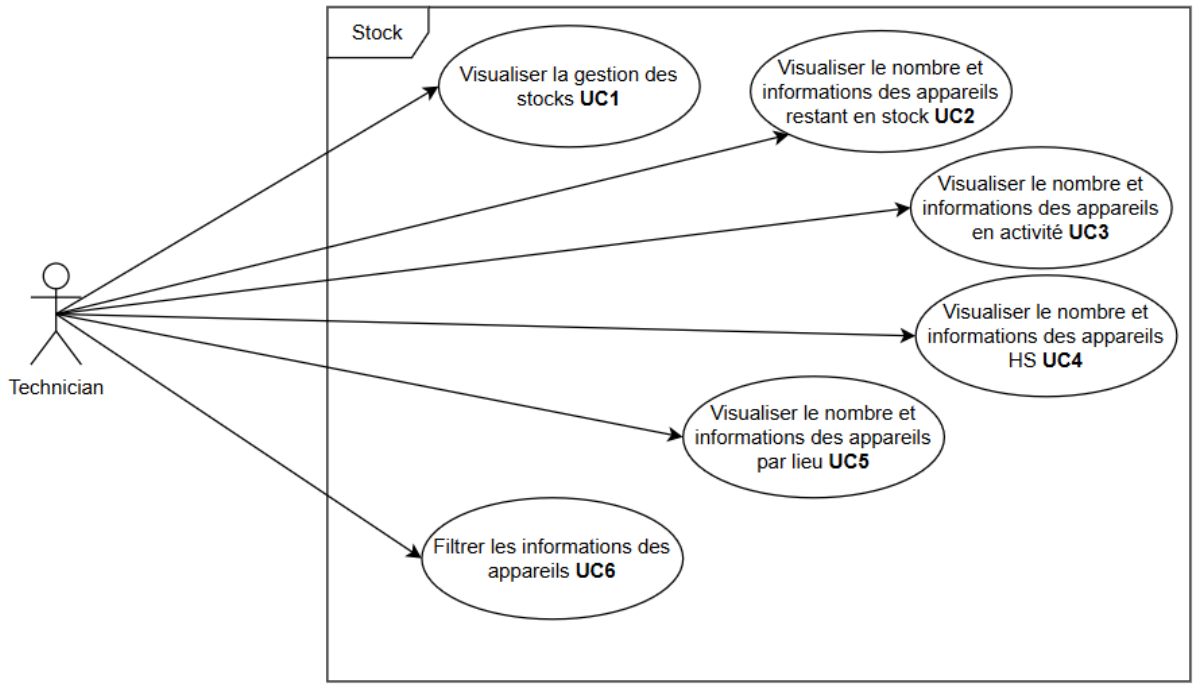
Accueil - GLPI

VIII. Annexe

VIII.3Diagramme de Gantt



VIII.4Diagramme Cas d'utilisation



VIII.5Annexe C : README

Évolution et amélioration de GLPI

FOR CCIAG CREATED BY JAROD TIVOLLIER

Ce projet a pour objectif de mettre à niveau le GLPI dans sa version la plus récente ainsi que ses plugins s'ils sont disponibles sur la version la plus récente et compatibles avec. Déterminer et remplacer l'existant par un plan de migration, Cela engendre de passer par une migration de données, d'ajouter des fonctionnalités telles que des nouveaux plugins en particulier un plugin de gestion des stocks qui est le besoin principal avec la montée de version de GLPI, des modifications des interfaces pour s'adapter aux critères de Bastien et Scapin (ensemble de critères ergonomiques pour l'évaluation d'interfaces utilisateur) et enfin vérifier le bon fonctionnement après toutes les modifications.

Pour commencer

Vous trouverez le dossier d'installation ici `/home/sysadmin/VM`

Prérequis

Ce projet a besoin de prérequis, que ce soit des connaissances ou des fichiers.

Pour ce qui est des connaissances :

- Docker
- Bash/Linux
- cron
- SQL

Pour ce qui est des fichiers, ils doivent tous être mis dans le dossier `/home/sysadmin/VM`

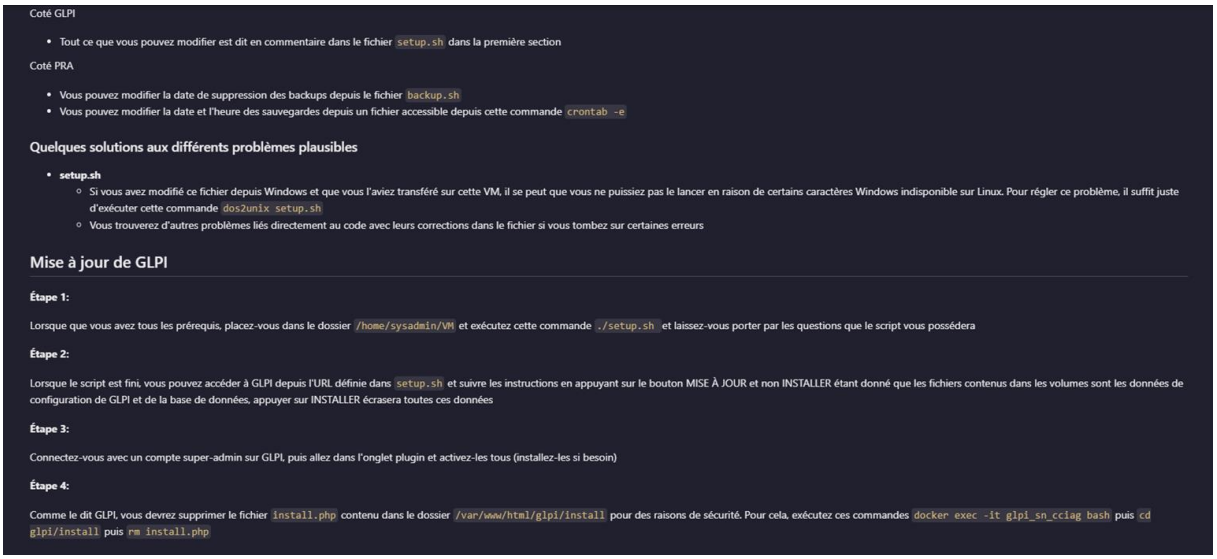
- **setup.sh** Ce fichier est le plus important il contient le code permettant la mise à jour semi-automatique de GLPI
- **docker-compose.yaml** Ce fichier contient les informations des conteneurs qui seront créés par le setup.sh
- **DockerfileDB** Ce fichier est un dockerfile, c'est un fichier d'installation de conteneur Docker. Ici, il crée un conteneur pour la base de données
- **glpicrypt.key** Ce fichier contient la clé d'activation pour le marketplace de GLPI
- **backup.sh** Ce fichier contient les instructions permettant de faire des backups des volumes Docker. Le fichier sera exécuté tous les certains temps depuis une tâche **cron**
- **README.md** Ce fichier contient les instructions pour faire la mise à jour GLPI

Configuration

Si vous souhaitez faire des modifications:

Coté base de données:

- Vous pouvez modifier le **nom de la base de données**, **l'utilisateur** et **son mot de passe** ainsi que **le mot de passe root** depuis le fichier `docker-compose.yaml`
- Si vous souhaitez ajouter directement une base de données qui remplace celle présente, regardez dans le fichier `DockerfileDB` pour plus d'informations



Coté GLPI

- Tout ce que vous pouvez modifier est dit en commentaire dans le fichier `setup.sh` dans la première section

Coté PRA

- Vous pouvez modifier la date de suppression des backups depuis le fichier `backup.sh`
- Vous pouvez modifier la date et l'heure des sauvegardes depuis un fichier accessible depuis cette commande `crontab -e`

Quelques solutions aux différents problèmes plausibles

- **setup.sh**
  - Si vous avez modifié ce fichier depuis Windows et que vous l'aviez transféré sur cette VM, il se peut que vous ne puissiez pas le lancer en raison de certains caractères Windows indisponible sur Linux. Pour régler ce problème, il suffit juste d'exécuter cette commande `dos2unix setup.sh`
  - Vous trouverez d'autres problèmes liés directement au code avec leurs corrections dans le fichier si vous tombez sur certaines erreurs

Mise à jour de GLPI

Étape 1:

Lorsque que vous avez tous les prérequis, placez-vous dans le dossier `/home/sysadmin/VM` et exécutez cette commande `./setup.sh` et laissez-vous porter par les questions que le script vous possédera

Étape 2:

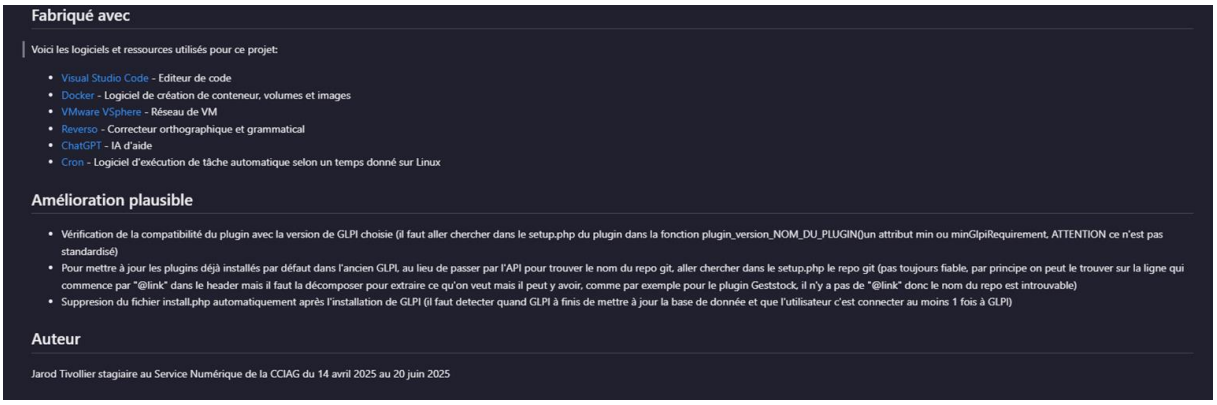
Lorsque le script est fini, vous pouvez accéder à GLPI depuis l'URL définie dans `setup.sh` et suivre les instructions en appuyant sur le bouton MISE À JOUR et non INSTALLER étant donné que les fichiers contenus dans les volumes sont les données de configuration de GLPI et de la base de données, appuyer sur INSTALLER écrasera toutes ces données

Étape 3:

Connectez-vous avec un compte super-admin sur GLPI, puis allez dans l'onglet plugin et activez-les tous (installez-les si besoin)

Étape 4:

Comme le dit GLPI, vous devrez supprimer le fichier `install.php` contenu dans le dossier `/var/www/html/glpi/install` pour des raisons de sécurité. Pour cela, exécutez ces commandes `docker exec -it glpi_sn_cciag bash` puis `cd glpi/install` puis `rm install.php`



Fabriqué avec

> Voici les logiciels et ressources utilisés pour ce projet:

- Visual Studio Code - Editeur de code
- Docker - Logiciel de création de conteneur, volumes et images
- VMware VSphere - Réseau de VM
- Reverso - Correcteur orthographique et grammatical
- ChatGPT - IA d'aide
- Cron - Logiciel d'exécution de tâche automatique selon un temps donné sur Linux

Amélioration plausible

- Vérification de la compatibilité du plugin avec la version de GLPI choisie (il faut aller chercher dans le setup.php du plugin dans la fonction plugin_version_NOM_DU_PLUGIN()un attribut min ou minGlpiRequirement. ATTENTION ce n'est pas standardisé)
- Pour mettre à jour les plugins déjà installés par défaut dans l'ancien GLPI, au lieu de passer par l'API pour trouver le nom du repo git, aller chercher dans le setup.php le repo git (pas toujours fiable, par principe on peut le trouver sur la ligne qui commence par "@link" dans le header mais il faut la décomposer pour extraire ce qu'on veut mais il peut y avoir, comme par exemple pour le plugin Geststock, il n'y a pas de "@link" donc le nom du repo est introuvable)
- Suppresion du fichier install.php automatiquement après l'installation de GLPI (il faut detecter quand GLPI à finis de mettre à jour la base de donnée et que l'utilisateur c'est connecter au moins 1 fois à GLPI)

Auteur

Jarod Tivollier stagiaire au Service Numérique de la CCIAG du 14 avril 2025 au 20 juin 2025