

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#### REPUBLIQUE DU BENIN

\*



MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

\*\*\*\*\*\*

DIRECTION GENERALE DE L'ENSEIGNEMENT SUPERIEUR

\*\*\*\*\*

**Institut Universitaire LES COURS SONOU** 

Filière : Informatique et Logiciel

# MEMOIRE DE FIN DE FORMATION POUR L'OBTENTION DU DIPLOME DE LICENCE PROFESSIONNELLE

#### THEME:

Développent d'une plateforme de supervision d'un réseau local : Cas de 2CL-Services

Réalisé et soutenu par

**BODJRENOU Ravène & BLENON Mardoché Charbel** 

Sous la supervision de :

Maître de stage : Maître de mémoire :

M. Corneille Lassymi ADJABA M. Abdel OLOUBO

Directeur du Centre 2CL-Services Ingénieur Logiciel

Année académique: 2022-2023

\*\*\*\*\*\*\*

## **ENGAGEMENT**

L'INSTITUT UNIVERSITAIRE LES COURS SONOU (LCS) N'ENTEND DONNER NI APPROBATION, NI IMPROBATION AUX OPINIONS EMISES DANS CE MEMOIRE. CES OPINIONS DOIVENT ETRE CONSIDEREES PROPRES A LEURS AUTEURS.

# DEDICACE ET REMERCIEMENTS

# **DEDICACE 1**

Je dédie ce travail à :

- \* Mon père, BODJRENOU François
- ❖ Ma mère, ASSOGBA Suzanne

Pour le soutien moral et financier ainsi que tous les efforts consentis pour ma formation et mon bienêtre.

BODJRENOU Y. G. Ravène

## **DEDICACE 2**

Je dédie ce travail à :

- \* Mon père, BLENON Edmond
- ❖ Ma mère, MOMBOUA LAETITIA

Pour le soutien moral et financier ainsi que tous les efforts consentis pour ma formation et mon bien-être.

BLENON Mardoché Charbel

# **REMERCIEMENTS**

Nous adressons nos remerciements à tous ceux qui, d'une manière ou d'une autre, ont contribué à la réalisation de ce mémoire et en l'occurrence :

- Au Président Directeur de l'institut Universitaire LES COURS SONOU (LCS), M. Fabrice SONOU et à ses collaborateurs;
- A notre Directeur des Etudes M. Gérard AGBIDINOUKOUN;
- ➤ A tous les enseignants de l'institut Universitaire LES COURS SONOU(LCS), pour les trois années d'éducation qu'ils nous ont donné, dans le but de nous booster ou de nous aider à acquérir plus de connaissance;
- ➤ A notre maitre de mémoire, **M. Abdel OLOUBO**, pour avoir accepté de superviser ce travail avec rigueur. Ses suggestions, remarques et encouragements nous ont été d'une grande importance ;
- ➢ A notre maitre de stage M. Corneille Lassymi ADJABA pour sa disponibilité, son soutien et ses encouragements ;
- A tous les étudiants de l'institut Universitaire LES COURS SONOU (LCS) Bohicon en particulier ceux de notre promotion, puis spécifiquement ceux de la filière informatique, pour le soutien accordé ainsi que pour la solidarité qui a toujours régné au milieu de nous.

# Sigles et Abréviations

Sigles et Abréviations	Désignation
CSS	
HTML	
JSON	
PHP	
UML	

# Listes des figures

Figure 1 : Type de diagramme UML	9
Figure 2 : Diagramme de cas d'utilisation	12
Figure 3 : Diagramme d'activité de cas d'utilisation inscription	16
Figure 4 : Diagramme d'activité de cas d'utilisation s'authentifier	17
Figure 5 : Diagramme de séquence cas d'utilisation s'inscrire	19
Figure 6 : Diagramme de séquence cas d'utilisation connexion	20
Figure 7 : Diagramme des classes	21
Figure 8 : Interface de connexion	26
Figure 9 : Interface d'inscription	27
Figure 10 : Interface de récupération du mot de passe	28
Figure 11 : Vérification de l'email	28
Figure 12 : Menu Principal	29
Figure 13 : Tableau de Bord	30
Figure 14 : Interface des services du serveur	30
Figure 15 : Interface de surveillance des machines	31
Figure 16 : Interface de contrôle de la connectivité	31
Figure 17 : Interface des machines connectées	32
Figure 18 : Interface des utilisateurs connectés	32
Figure 19 : Interface du serveur	33
Figure 20 : Capture du code de création du compte	34
Figure 21 : Capture du code d'enregistrement des machines	35
Figure 22 : Capture du code de la mise à jour du profil	36
Figure 23 : Capture du code des routes	37

# Listes des tableaux

Tableau 1 : Identification des acteurs et leurs rôles	. 11
Tableau 2 : Dictionnaire des données	22

# Sommaire

INTRODUCTION	1
PREMIERE PARTIE: PRESENTATION DU CADRE ET DU SUJET D'ETUDE	2
CHAPITRE 1 : PRESENTATION DE LA STRUCTURE D'ACCUEIL	3
CHAPITRE 2 : PRESENTATION DE LA SOLUTION A CONCEVOIR	4
DEUXIEME PARTIE : ANALYSE ET CONCEPTION DU SYSTEME	7
CHAPITRE 3: METHODOLOGIE	8
CHAPITRE 4: CONCEPTION DU SYSTEME	23
CONCLUSION ET PERSPECTIVES	38

# Résumé

Le présent mémoire résume notre projet de fin d'études dans le cadre de l'obtention du diplôme de licence professionnelle en Informatique option Système Informatique et Logiciel. Ce projet consiste à la réalisation d'une plateforme de supervision d'un réseau local. Il a pour but de superviser à distance le serveur et d'autres matériels informatique associés, tous connectés dans un même réseau afin de prévenir les disfonctionnements d'un matériel connecté au réseau. La réalisation de ce projet a nécessité l'utilisation de plusieurs outils de travail. Nous nous sommes servis d'UML comme outil d'analyse et de modélisation, de Visual studio code comme éditeur de code extensible ; Php et Javascript comme languages de programmation ; Laravel et Bootstrap comme frameworks ; MySQL comme SGBD. Tout ce qui précède concourt à la portabilité, la maintenance et l'évolutivité de cette solution.

<u>Mots clés:</u> Laravel, Bootstrap, Php, Javascript, UML, MySQL, SGBD, Portabilité, Maintenance, Evolutivité.

# **Abstract**

This thesis summarizes our end-of-studies project in the context of obtaining the professional bachelor's degree in computer science, computer system and software option. This project consists in the realization of a monitoring platform of a local network. Its purpose is to remotely supervise the server and other associated computer equipment, all connected in the same network in order to prevent malfunctions of equipment connected to the network. The realization of this project required the use of several work tools. We used UML as an analysis and modeling tool, Visual studio code as an extensible code editor; Php and JavaScript as programming languages; Laravel and Bootstrap as frameworks; MySQL as DBMS. All of the above contributes to the portability, maintainability and scalability of this solution.

# **INTRODUCTION**

Les notions de sciences et de techniques se transforment à un rythme rapide depuis la grande révolution scientifique du 18<sup>ème</sup> siècle. Ces changements ont apporté de nouvelles satisfactions, une autonomie renouvelée et une forte dose d'innovation. Mais de nos jours l'on parle plutôt des nouvelles technologies de communication et de l'information avec l'informatique qui devient de plus en plus incontournable. Science du traitement rationnel de l'information, l'informatique s'impose plus que jamais, comme un outil précieux dont a besoin toute entreprise qui veut accroître sa productivité et rester compétitive sur le plan international. Un tel outil aide à bien satisfaire les exigences croissantes non seulement des administrateurs, mais aussi des utilisateurs. En effet, toutes les entreprises, de nos jours, devraient être équipées d'un réseau local au minimum et l'apparition de ces nouveaux environnements informatisés rend la surveillance des éléments clefs du réseau et du système une opération indispensable, afin de minimiser la perte d'exploitation et garantir que les utilisateurs ne s'aperçoivent pas des anomalies de fonctionnement. Vu que le système informatique est au cœur des activités d'entreprise, sa maîtrise devient primordiale, puisque, il doit fonctionner pleinement et en permanence pour garantir la fiabilité et l'efficacité exigées, d'une part. D'autre part, les problèmes liés au système informatique tels que les défaillances, les pannes, les coupures et les différents problèmes techniques doivent être réduits, du fait qu'une indisponibilité du système ou du réseau peut causer des pertes considérables. Afin de minimiser le nombre de ces pertes et d'accélérer la maintenance, une sorte de surveillance et de contrôle s'avère obligatoire. Ainsi le lien étroit existant entre la supervision du parc informatique d'une entreprise et sa productivité suscite notre curiosité et justifie certainement le choix de notre thème : << Développement d'une plateforme de supervision d'un réseau local : Cas de 2CL-**Services** >> pour pallier aux problèmes rencontrés durant notre stage.

# PREMIERE PARTIE: PRESENTATION DU CADRE ET DU SUJET D'ETUDE

# CHAPITRE 1 : PRESENTATION DE LA STRUCTURE D'ACCUEIL

# CHAPITRE 2 : PRESENTATION DE LA SOLUTION A CONCEVOIR

Pour réaliser le présent travail, il est important d'effectuer une collecte d'information afin de prendre connaissance des travaux effectués et des solutions existantes s'intégrant dans le même sens. Dans cette partie, il s'agira de faire une synthèse sur l'existant.

# 2\_1- Etude de l'existant

Le parc informatique de 2CL-Services est constitué des appareils suivants :

- 33 ordinateurs dont 30 dans la salle de formation et 03 dans le bureau administratif
- Un serveur VoIP qui gère les appels internes de l'entreprise
- Un serveur proxy constitué de pfsense et de squid pour contrôler et filtrer l'accès à
   l'internet à partir des adresses MACs
- Une imprimante
- Une photocopieuse
- Un Switch à 24 ports
- 1 routeur de bord ayant 04 ports Ethernet et une carte wifi
- Un pont wifi pour relayer le signal du routeur au niveau de la salle de formation
- o 1 modem
- 20 prises murales Ethernet RJ45
- 02 prises murales téléphonique RJ11
- 02 téléphones IP et les portables Android des personnels du centre

Les machines du parc informatique de 2CL-Services utilisent le système d'exploitation linux distribution Ubuntu avec une machine virtuelle (Virtual box 7.0.1) sur laquelle est installé le système d'exploitation Windows 10. Les logiciels les plus utilisés sur les machines de la salle de formation sont les logiciels utilisés pour former les apprenants tels que Microsoft Office (Word, Excel, Access, Publisher, Power Point, etc...), Microsoft Project pour les formations de gestion de projet, Perfect, Adobe Collector, etc. Tous les instruments électriques du centre sont protégés par un régulateur. Ces machines travaillent en réseau.

# Critique de l'existants

Le système en place fonctionne mais sans un outil de supervision. La maintenance curative s'effectue par isolement jusqu'à détecter l'endroit précis du disfonctionnement. Le centre ne dispose aucune plateforme pour diagnostiquer en temps réel l'état des machines y compris les serveurs.

# 2\_2- Problématique

Avec le système existant, il y a l'inefficacité de détection des pannes, de contrôle des disques dur de stockage des serveurs et du contrôle de la connectivité des personnels et des machines. Il n'y a pas un outil de supervision du réseau local dans le centre.

# 2\_3 - Solution Proposées

Afin de résoudre ces problèmes, le centre se tourne vers une solution digitale. La conception d'une plateforme de supervision du réseau local dont le but est de faciliter la tâche aux administrateurs, pour assurer la gestion et la supervision des équipements du centre avec efficacité.

# 2<sub>4</sub> - Objectifs

L'objectif visé à travers la conception de cette plateforme pour le centre 2CL-Services est de faciliter la gestion et la supervision des équipements. De façon spécifique notre plateforme permet :

- d'éteindre et de redémarrer le serveur à distance ;
- de surveiller les services du serveur à distance ;
- o d'éteindre et de redémarrer les ordinateurs connectés au réseau à distance ;
- de surveillé la connectivité des appareils dans le réseau ;
- o de surveillez l'action des administrateurs connectés depuis la plateforme ;
- de gérer les droits des utilisateurs inscrits sur la plateforme et de tracer leurs connectivités.

# **2\_5 – Limites**

Les limites de notre plateforme sont :

- o Elle n'est pas en ligne ;
- o Elle ne supervise pas les câbles RJ45 et RJ11;
- o Elle ne contrôle pas l'espace disque dur des machines connectées.

# DEUXIEME PARTIE : ANALYSE ET CONCEPTION DU SYSTEME

# **CHAPITRE 3: METHODOLOGIE**

Dans la conception d'un système d'information, la modélisation des données est l'analyse et la conception de l'information contenue dans le système afin de représenter la structure de ces informations et de structurer le stockage et les traitements informatiques. Nous disposons de plusieurs méthodes de modélisation, parmi lesquelles, nous avons :

#### **UML (Unified Modeling Language)**

C'est clairement le standard de référence, la technique la plus connue et enseignée à ce jour. Il est vrai qu'elle permet de visualiser et de conceptualiser à peu près toutes les facettes d'une demande de changement, et qu'elle est compréhensible autant par les interlocuteurs métier que technique. En business analysis, cette boîte à outils permet de modéliser les cas d'utilisation, les fonctionnalités, les processus, de représenter les interactions entre composants, acteurs, de préparer la stratégie de déploiement de la solution cible... Bien entendu, il faut en apprendre la syntaxe, mais même sans l'appliquer stricto sensu, ses diagrammes et vues permettent au Business Analyst d'analyser et de rédiger la plupart de ses livrables.

#### **MERISE**

Merise est une méthode séquentielle, par opposition aux méthodes itératives et incrémentales. Cette technique est très intéressante pour analyser, concevoir et réaliser des systèmes d'information. Il faut bien entendu en apprendre la syntaxe, tout comme l'UML, mais sa maîtrise en fait un outil puissant et rigoureux si elle est partagée entre développeurs et Business Analysts. Cependant, elle est de plus en plus rarement enseignée en école d'ingénieur ou en master informatique, au profit de l'UML, et sa « lourdeur » ne se prête pas aux projets agiles.

Nous optons donc pour l'UML qui est un langage formel, normalisé (gage de stabilité, encourage l'utilisateur d'outils) et un support de communication performant

#### 3.1 Présentation des méthodes de modélisation



UML se définit comme un langage de Modélisation Unifié, en anglais Unified Modeling Language, se présente comme le langage de modélisation graphique à base de pictogrammes conçu comme une

méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orienté objet. Ce langage est constitué de digrammes intégrés par les développeurs informatiques pour la représentation visuelle des objets, des états et des processus dans un logiciel ou dans un système. Cependant UML intéresse fortement les spécialistes de l'ingénierie système. UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis la spécification jusqu'au codage. Dans ce cadre, un concept appartenant aux besoins des utilisateurs projette sa réalité dans le modèle de conception et dans le codage.

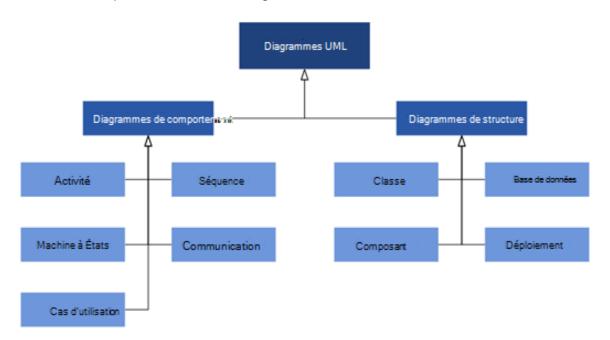


Figure 1 : Type de diagramme UML

### 3-1-1- Présentation des diagrammes de comportement

## 1- Diagramme des cas d'utilisation

### **❖ Identification des acteurs**

Un acteur d'un système représente une entité externe à ce système qui interagit directement (saisie de données, réception d'information...) avec lui. On trouve les acteurs

en observant les utilisateurs directs du système, les responsables de la maintenance, ainsi que les autres systèmes qui interagissent avec lui. Pour notre système, nous avons comme acteurs:

- Les personnels du service informatique
- L'administrateur
- Le super administrateur

Acteurs	Cas d'utilisation
	Tous les cas d'utilisation
	d'administrateur SI
	S'authentifier
Le Super Admin	Consulter les statistiques de connexion
	Administrer:
	<ul><li>Nommer un administrateur</li><li>Révoquer un administrateur</li></ul>
	Gérer la journalisation de connexion :
	<ul> <li>Supprimer le journal de</li> </ul>
	connexion des utilisateurs et des machines

Acteurs	Cas d'utilisation		
L'administrateur	Tous les cas d'utilisation du personnel SI  Gérer les personnels SI: - Supprimer un personnel SI sauf le super admin  Gérer les matériels: - Ajouter un matériel au réseau - Supprimer un matériel  Gérer le serveur: - Eteindre le serveur - Redémarrer le serveur  Gérer les ordinateurs connectés: - Eteindre un ordinateur connecté au réseau - Redémarrer un ordinateur connecté		
Les personnels SI	- Redémarrer un ordinateur connecte au réseau  Gérer son compte : - Créer un compte - Vérifier l'email - Se connecter - Changer sa photo de profil - Modifier ses informations - Modifier son mot de passe  Consulter les différents utilisateurs  Consulter les différents matériels  Consulter l'état des différents services du serveur  Consulter l'état du disque dur du serveur  Consulter l'état du réseau		

Tableau 1 : Identification des acteurs et leurs rôles

# ❖ Élaboration du diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation décrivent les interactions entre le système et les acteurs. Détaillons quelques cas d'utilisation avec leurs scénarios. Chaque scénario est composé d'étape. Une étape est un message de l'acteur vers le système, une validation ou changement d'état du système ou encore un message du système vers un acteur. Un acteur et un cas d'utilisation sont mis en relation par une association représentée par une ligne.

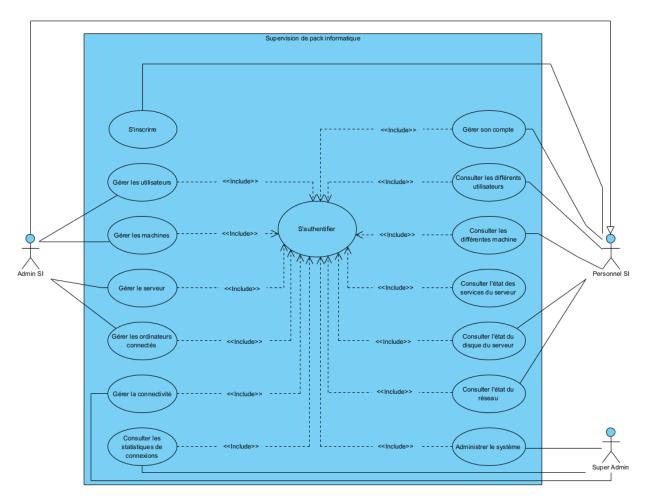


Figure 2 : Diagramme de cas d'utilisation

Description textuelle du diagramme des cas d'utilisation

Cas d'utilisation « S'inscrire »

Titre: S'inscrire

**Résumé**: C'est la démarche à effectuer par l'utilisateur du système l'obligeant à s'inscrire avant de pouvoir se connecter les fois à venir.

**Acteurs**: Personnel SI, Admin SI.

#### Description du scénario

Ce cas commence lorsque l'utilisateur veut intégrer le système pour la toute première fois. Le système affiche la page d'authentification et demande à l'utilisateur de s'inscrire s'il n'a pas de compte. L'utilisateur accède à l'interface d'inscription et renseigne ses informations. Le système vérifie si l'email utilisé par l'utilisateur a été déjà utilisé pour une ancienne inscription et rejette si oui et si non envoie un email de vérification. L'utilisateur vérifie son email.

Le système donne l'accès à l'interface de la plateforme.

#### **Préconditions:**

Être un utilisateur et être capable de s'inscrire en renseignant les informations demandées.

S'inscrire depuis un appareil enregistré dans le réseau.

#### **Enchainement nominal**

- 1-L'utilisateur renseigne le lien de la plateforme dans la barre de recherche de son navigateur.
- 2-La page de connexion s'ouvre en premier.
- 3- L'utilisateur clique sur S'inscrire en bas du bouton Se connecter.
- 4-L'utilisateur renseigne les informations demandées par le formulaire.
- 5- Le système vérifie si l'email a été déjà utilisé.
- 6-Le système envoie une vérification d'email à l'utilisateur.
- 7-L'utilisateur vérifie l'email
- 8-Le système redirige l'utilisateur sur la page home et enregistre sa le journal de connexion.

**Alternative** : Si l'appareil utilisé par l'utilisateur pour l'inscription n'est pas enregistré dans la base de données, le système n'affiche pas la page d'inscription et renvoie une page 404.

Si un champ d'information n'est pas valide ou si l'email a été déjà utilisé, le Système affiche un message d'erreur et réaffiche la page d'inscription.

#### Cas d'utilisation « S'authentifier »

#### Titre: S'authentifier

**Résumé** : C'est la démarche à effectuer par l'utilisateur du système l'obligeant à s'authentifier avant d'avoir accès à la fonctionnalité liée à sa section.

**Acteurs**: Personnel SI, Admin SI, Super Admin

#### **Description du scénario**

Ce cas commence lorsque l'utilisateur veut avoir accès à sa zone de travail. Le système affiche l'interface d'authentification. Le système vérifie l'existence de l'utilisateur et de la machine utilisée pour l'authentification.

Le système donne l'accès à l'interface de la plateforme

#### **Préconditions:**

Être un utilisateur disposant d'un email et un mot de passe.

Se connecter depuis un appareil enregistré dans le réseau

#### **Enchainement nominal**

- 1-L'utilisateur renseigne de lien de la plateforme dans la barre de recherche de son navigateur.
- 2-La page de connexion s'ouvre en premier.
- 3- L'utilisateur saisi son email et son mot de passe.
- 4-L'utilisateur valide en appuyant sur le bouton **Se connecter**.
- 5- Le système vérifie la conformité de l'email et du mot de passe

6-Le système redirige l'utilisateur sur la page home et enregistre son journal de connexion.

**Alternative** : Si l'appareil utilisé par l'utilisateur pour la connexion n'est pas enregistré dans la base de données, le système n'affiche pas la page de connexion et renvoie une page 404.

Si un champ d'information n'est pas valide ou si l'utilisateur n'existe pas, le système affiche un message d'erreur et réaffiche la page de connexion.

#### Cas d'utilisation « Gérer le serveur »

Titre: Gérer le serveur

Résumé : C'est la démarche permettant d'éteindre et de rallumer le serveur

Acteur: Admin SI et Super admin

#### Description du scénario :

- 1- Pour éteindre le serveur l'Admin SI ou le Super admin clique sur l'icône **power** et le système demande s'il veut vraiment éteindre et il confirme l'action et le serveur s'éteint
- 2- Pour redémarrer le serveur l'Admin SI ou le Super admin clique sur **restart** et le système demande s'il veut vraiment redémarrer et il confirme l'action et le serveur redémarre

**Précondition** : l'Admin SI ou le Super admin est connecté à la plateforme.

#### **Enchainement nominal**

- 1-L'Admin ou le Super admin se connecte à la plateforme
- 2-L'Admin SI ou le Super admin va sur *Outil* ensuite sur *Gérer le serveur*
- 3-L'Admin ou le Super admin clique sur l'icône *power* ou *restart* pour éteindre ou redémarrer le serveur

# 2- Diagramme d'activités

Le diagramme d'activités représente les activités que réalisent un ou plusieurs objets. Il peut correspondre à la description en détail d'une activité du diagramme d'états transitions, à la description d'une méthode. Il peut également décrire l'activité d'un

système ou d'un sous-système en assignant les responsabilités à chaque acteur. Le diagramme d'activités constitue aussi un bon choix pour décrire un cas d'utilisation.

### Diagramme d'activité du cas « Inscription »

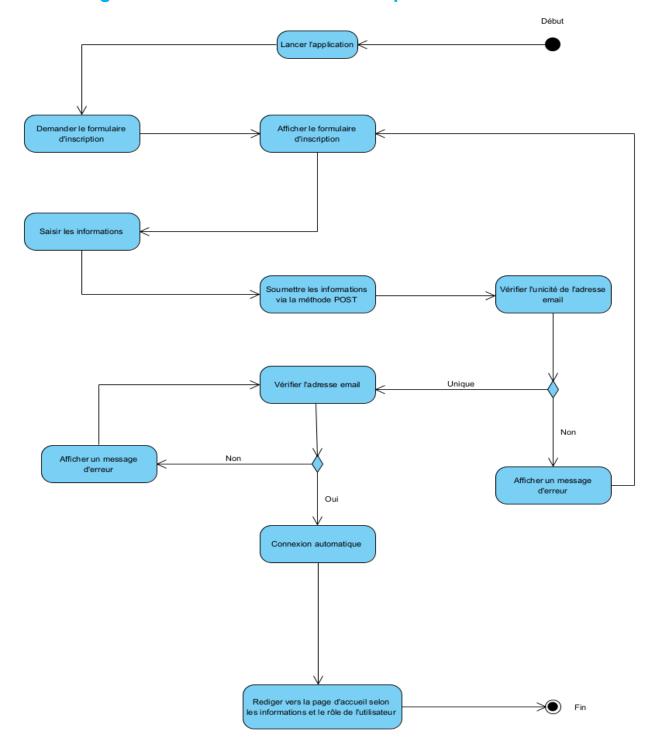


Figure 3 : Diagramme d'activité de cas d'utilisation inscription

# Diagramme d'activité du cas « connexion »

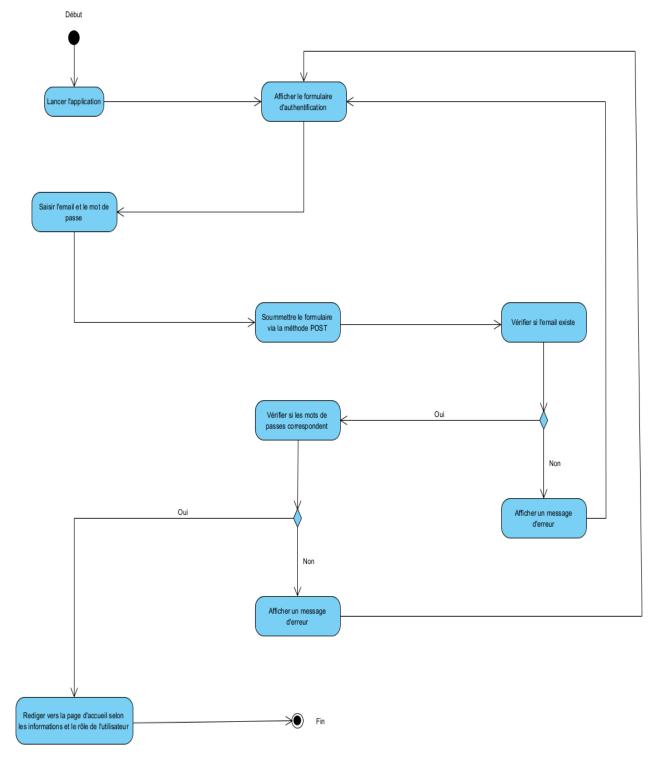


Figure 4 : Diagramme d'activité de cas d'utilisation s'authentifier

## 3-1-2- Présentation du Diagramme d'interaction (Séquence)

Le diagramme de séquence décrit la dynamique du système. À moins de modéliser un très petit système, il est difficile de représenter toute la dynamique d'un système sur un seul diagramme. Aussi la dynamique globale sera représentée par un ensemble de diagrammes de séquence, chacun étant généralement lié à une sous fonction du système. Le diagramme de séquence décrit les interactions entre un groupe d'objets en montrant, de façon séquentielle, les envois de message qui interviennent entre les objets. Le diagramme peut également montrer les flux de données échangées lors des envois de message.

Diagramme de Sequence << inscription >>

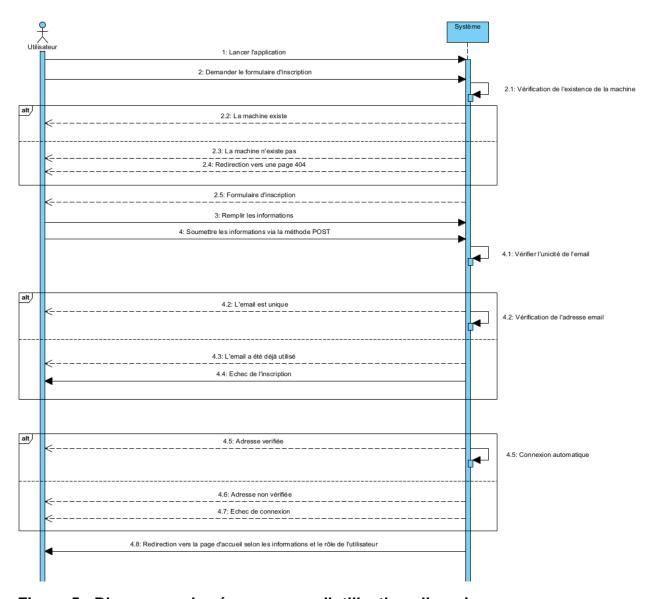


Figure 5 : Diagramme de séquence cas d'utilisation s'inscrire

Diagramme de Sequence << connexion >>

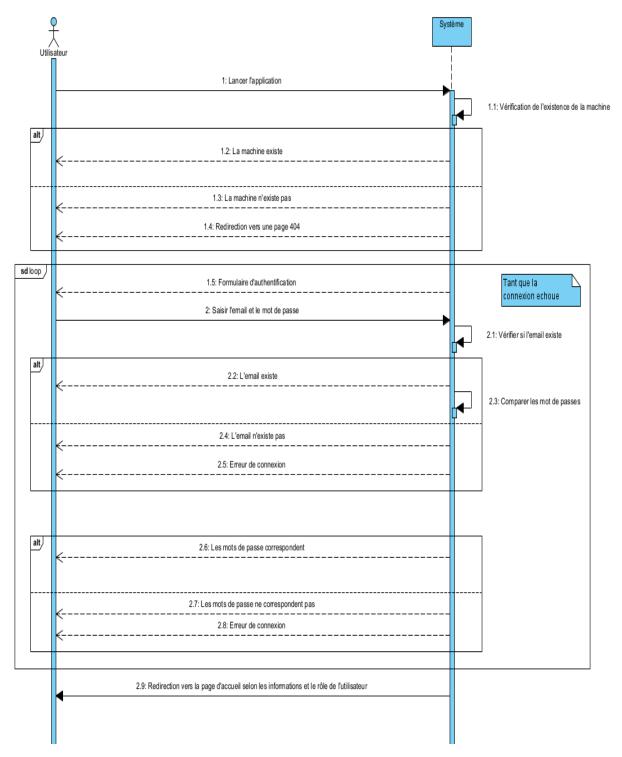


Figure 6 : Diagramme de séquence cas d'utilisation connexion

## 3\_1\_3- Présentation du Diagramme Structurel (Classe)

## 1. Diagramme de classes

Le diagramme de classe de conception représente bien la structure statique du code, par le biais des attributs et des relations entre classes. C'est un diagramme principal qui est la vue du plus haut niveau avec l'ensemble des classes de l'application. Il montre les briques de base statiques : classes, associations, interfaces, attributs, opérations, généralisations, etc. On peut avoir des vues logiques plus spécialisées si nécessaire :

- Vue des classes participant à un scénario
- Vue des classes "privées" dans un package
- Vue d'une classe avec ses attributs et ses opérations
- Vue d'une hiérarchie d'héritage

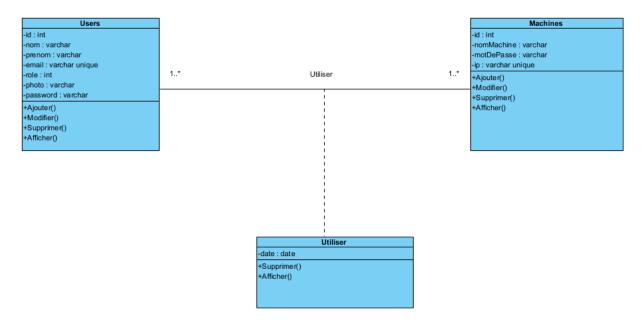


Figure 7 : Diagramme des classes

Description détaillée du diagramme de classes

#### Règle de domaine

- Chaque Utilisateur est caractérisé par un id, nom, prénom, email, rôle, photo, password.

- Chaque Machine est caractérisée par un id, nomMachine, motDePasse, IP.
- Un utilisateur n'a qu'un seul compte.
- L'adresse IP est spécifique pour chaque machine.
- Un utilisateur utilise au moins une machine dans le réseau.
- Une machine du réseau sert au moins un utilisateur.

### 2. Dictionnaire de données

Propriété	Désignation	Туре	Taille	Entités
id	Identifiant de l'entité	Numérique	-	Toutes les classes
nom	Nom de l'utilisateur	Alphanumérique	255	Users
prenom	Prénom de l'utilisateur	Alphanumérique	255	Users
role	Rôle de l'utilisateur	Numérique	-	Users
password	Mot de passe de l'utilisateur	Alphanumérique	255	Users
id	Identifiant de la machine	Numérique	-	Machines
nomMachine	Nom de la machine	Alphanumérique	255	Machines
motDePasse	Mot de passe de la machine	Alphanumérique	255	Machines
ip	Adresse ip de la machine	Alphanumérique	255	Machines

### Tableau 2 : Dictionnaire des données

# 3. Modèle Logique Relationnel

Users (id, nom, prenom, email, role, photo, password)

Machine (id, nomMachine, motDePasse, ip)

Utiliser (#idUsers,#idMachine, date)

# **CHAPITRE 4: CONCEPTION DU SYSTEME**

# 4\_1- Présentation des outils de développement

## **4\_1\_1-** Les Langages de développement

# **❖ Languages de programmation**

#### > PHP



PHP (HyperText Preprocessor) est un langage de programmation libre, principalement utilisé pour produire des pages web dynamiques via un serveur http. C'est un langage impératif orienté objet.

#### > JAVASCRIPT



JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web.

# 0

#### > JSON

JSON (Javascript Object Notation) est un langage léger d'échange de données textuelles. C'est un format de données textuelles dérivé de la notation des objets du langage Javascript.

### \*Autres langages

# HTML

#### > HTML

HTML signifie « HyperText Markup Language » qu'on peut traduire par langage de balises pour l'hypertexte. C'est un langage de balisage qui sert à indiquer au navigateur comment structurer les pages web visitées.

#### > CSS



Les feuilles de style en cascade, généralement appelées CSS de l'anglais « Cascading Style Sheets », forment un langage informatique qui décrit la présentation des documents HTML et XML.

#### Framework

#### LARAVEL



Laravel a été créé par Taylor Otwell en juin 2011.

Le référentiel Laravel/laravel présent sur le site GitHub contient le code source des premières versions de Laravel. À partir de la cinquième version, le framework est développé au sein du référentiel Laravel/framework.

Laravel reste pourtant basé sur son grand frère Symfony, pour au moins 30 % de ses lignes (utilisation de "Symfony component").

La version 10 de Laravel nécessite au minimum PHP 8.1 et son installation est basée sur le gestionnaire de paquets Composer.

Laravel fournit des fonctionnalités en termes de routage de requête, de mapping objetrelationnel (un système baptisé Eloquent implémentant Active Record), d'authentification, de vue (avec Blade), de migration de base de données, de gestion des exceptions et de test unitaire.

#### **BOOTSTRAP**



Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc...), de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

#### \* SGBD

#### > MYSQL



MySQL est un système de gestion de bases de données relationnelle open source. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde.

#### **❖ SERVER MONITOR**

#### Laravel serveur monitor



Package Laravel pour surveiller périodiquement la santé de votre serveur et de votre application. Il est livré avec des vérifications communes prêtes à l'emploi et vous permet également d'ajouter vos

propres vérifications personnalisées. Les packages sont livrés avec des interfaces console et Web PHP >= 8+ et Laravel 9+.

Le paquet est livré avec les vérifications suivantes prêtes à l'emploi. Les chèques peuvent être divisés en trois catégories :

Contrôles de serveur : contrôles liés uniquement à votre serveur.

Vérifications communes : vérifications qui concernent uniquement votre application, mais qui sont de nature courante, quel que soit l'environnement dans lequel votre application s'exécute. Ces vérifications s'exécutent sur tous les environnements.

Contrôles d'environnement : contrôles liés à votre application uniquement, mais limités à un environnement spécifique tel que la production ou le développement.

## 4\_1\_2- Les environnements de développement

#### > Visual studio code



Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS.

# 4\_2- Présentation de l'application

## **4\_2\_1- Quelques Interfaces De Notre Projet**

#### 1. Interface de connexion

Première interface du logiciel, elle permet de se connecter au logiciel avec l'email et le mot de passe.

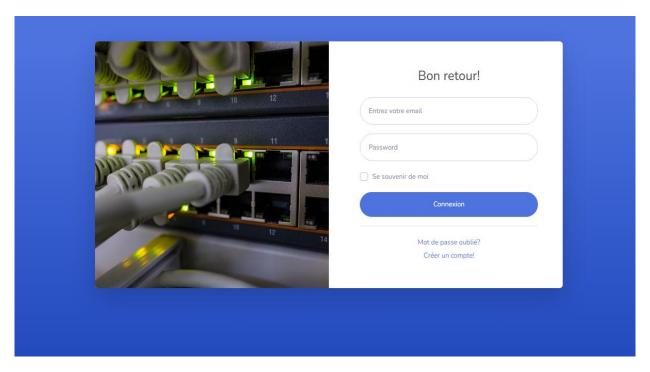


Figure 8 : Interface de connexion

## 2. Interface d'inscription

Interface de création de compte avec un mot de passe sécurisé et une adresse email existante et unique.

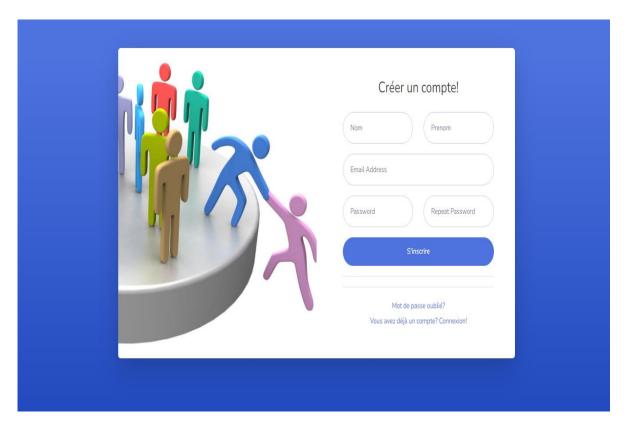


Figure 9: Interface d'inscription

## 3. Interface de récupération du mot de passe

Interface de récupération du mot de passe avec une adresse email existante dans la base de données.



Figure 10 : Interface de récupération du mot de passe

## 4. Vérification de l'email

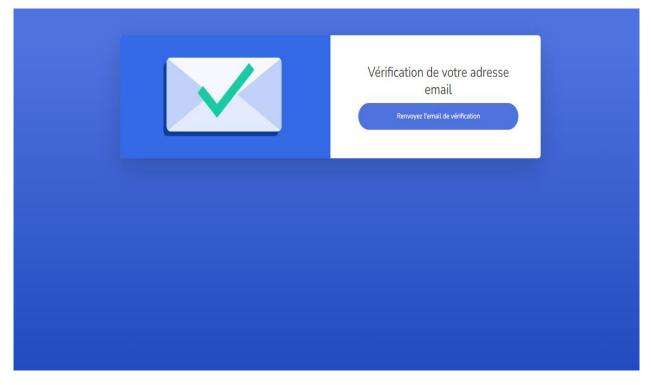


Figure 11 : Vérification de l'email

## 5. Menu Principal

Le menu principal de la plateforme :



Figure 12: Menu Principal

#### 6. Tableau de bord

Le tableau de bord présente une vue statistique sur le nombre d'utilisateur et de machines enregistré en fonction d'Adresse IP et sur le journal de connexion par utilisateur d'une part et d'autre part par machine.

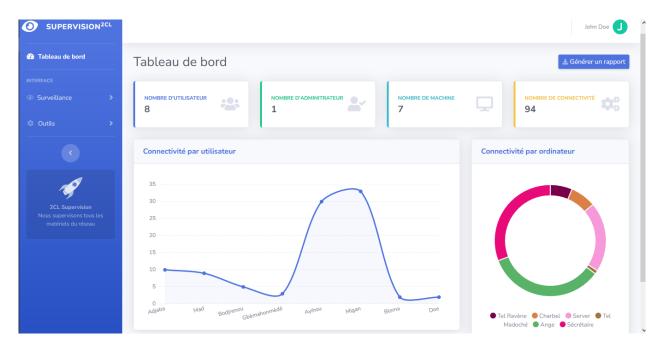


Figure 13: Tableau de Bord

#### 7. Interface des services du serveur

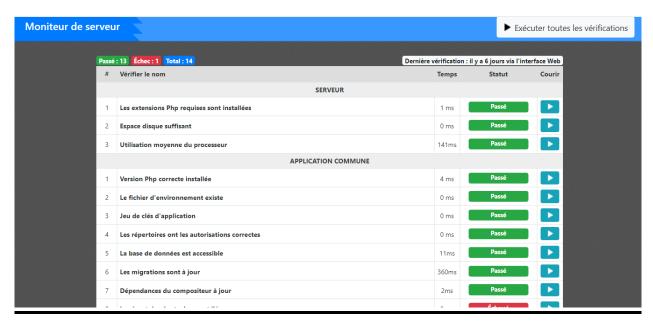


Figure 14 : Interface des services du serveur

#### 8. Interface de surveillance des machines

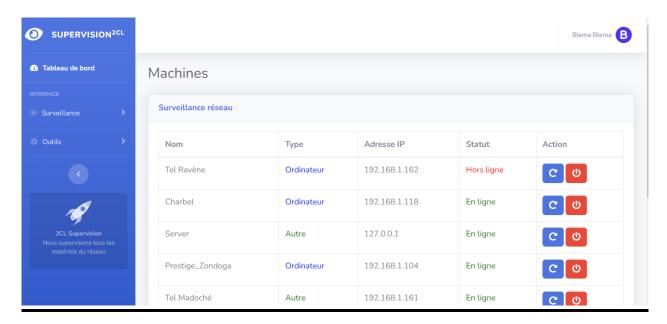


Figure 15 : Interface de surveillance des machines

### 9. Interface du journal de connexion

Cette interface permet de voir les différentes connexions par utilisateur et par machines.

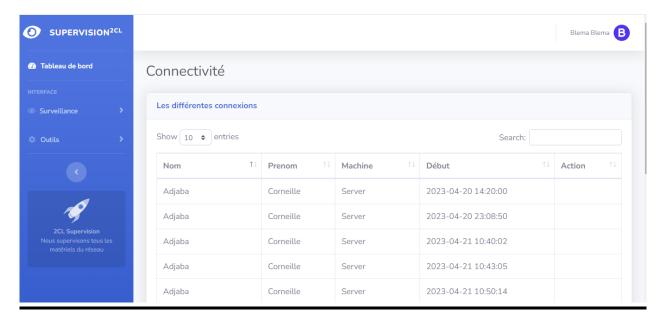


Figure 16 : Interface de contrôle de la connectivité

#### 10. Interface des machines connectées au serveur

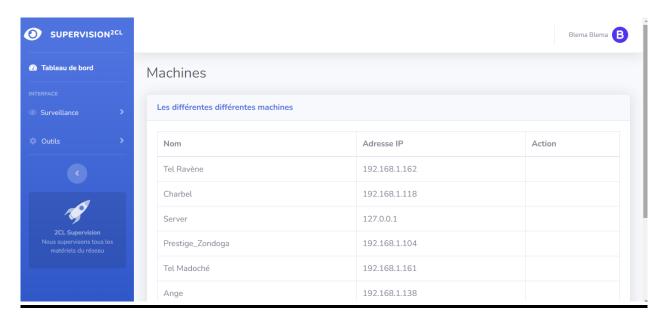


Figure 17 : Interface des machines connectées

## 11. Interface des personnels SI

Cette interface permet de montrer ceux qui se sont inscrit sur la plateforme de supervision.

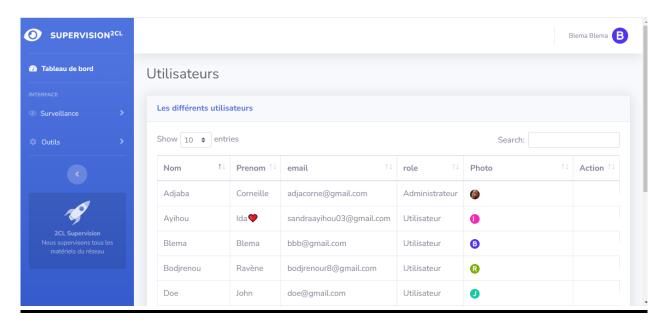


Figure 18 : Interface des utilisateurs connectés

## 12. Interface de gestion du serveur

lci on peut redémarrer le serveur et l'éteindre et voir aussi l'évolution de l'espace disque

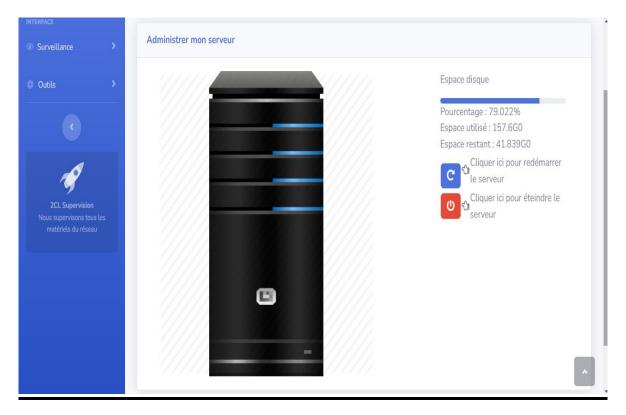


Figure 19 : Interface du serveur

# **4\_2\_2- Quelques Code Source De La Réalisation**

# 1. Code de Création de Compte

```
20
         public function create(array $input): User
21
22
             Validator::make($input, [
23
                  'nom' => ['required', 'string', 'max:255'],
                  'prenom' => ['required', 'string', 'max:255'],
24
25
                  'email' => [
26
                      'required',
                      'string',
27
28
                      'email',
29
                      'max:255',
30
                      Rule::unique(User::class),
31
32
                  'password' => $this->passwordRules(),
33
             ])->validate();
34
35
36
              * Make Avatar
37
38
39
             $path = 'users/images/';
             $fontPath = public_path('fonts/Oliciy.ttf');
41
             $char = strtoupper($input['prenom'][0]);
             $newAvatarName = rand(12, 34353) . time() . '_avatar.png';
42
43
             $dest = $path . $newAvatarName;
44
45
             $createAvatar = makeAvatar($fontPath, $dest, $char);
             $picture = $createAvatar == true ? $newAvatarName : '';
48
             return User::create([
49
                 'nom' => $input['nom'],
50
                  'prenom' => $input['prenom'],
51
                  'email' => $input['email'],
                 'role' => 0,
                 'photo' => $picture,
53
                  'password' => Hash::make($input['password']),
54
55
             1);
56
57
```

Figure 20 : Capture du code de création du compte

### 2. Code d'enregistrement des machines

```
class AdminController extends Controller
21
22
         public function index()
23
24
25
             if (Auth::user()->role == 2) {
26
                 $utilisateurs = User::all();
                 $nb_utilisateurs = $utilisateurs->count();
27
28
                 $nb_administrateurs = User::where('role', '1')->count();
                 //$nb_administrateurs = $administrateurs->count();
29
                 $machines = Machine::all();
30
31
                 $nb_machines = $machines->count();
32
                 $connexions = Utiliser::all();
33
                 $nb_connexions = $connexions->count();
34
                 return view('dashboard', [
                     "title" => 'Tableau de bord',
35
                     "nb_utilisateurs" => $nb_utilisateurs,
36
                     "nb_administrateurs" => $nb_administrateurs,
37
38
                     "nb_machines" => $nb_machines,
39
                     "utilisateurs" => $utilisateurs,
40
                     "machines" => $machines,
                     "nb_connexions" => $nb_connexions
41
42
                 1);
43
             } else {
44
                 return view('home', [
45
                     'title' => 'Accueil',
                      'sub-title' => 'Accueil',
47
                 1);
48
49
```

Figure 21 : Capture du code d'enregistrement des machines

### 3. Code de changement de profil

```
114
          public function updateProfile(Request $request)
115
116
117
118
              $validator = FacadesValidator::make($request->all(), [
119
                 'nom' => 'required',
                  'prenom' => 'required',
120
121
                 'email' => 'required|email'
122
              ]);
123
124
              if ($validator->fails()) {
125
                  return response()->json(['status' => 0, 'error' => $validator->errors()->toArray()]);
126
              } else {
127
                  if (Auth::user()->email == $request->email) {
128
                      $query = User::find(Auth::user()->id)->forceFill([
129
                          'nom' => $request->nom,
130
                          'prenom' => $request->prenom,
                      ])->save();
131
132
                  } else {
133
                     $query = User::find(Auth::user()->id)->forceFill([
134
                          'nom' => $request->nom,
135
                          'prenom' => $request->prenom,
                          'email' => $request->email,
136
137
                         'email_verified_at' => null
138
                      ])->save();
139
140
141
                  if (!$query) {
                      return response()->json(['status' => 0, 'msg' => 'Echec de modification des informations']);
143
                  } else {
                      return response()->json(['status' => 1, 'msg' => 'Vos informations ont été bien modifiée']);
144
145
147
```

Figure 22 : Capture du code de la mise à jour du profil

#### 4. Code des routes

```
Route::get('/', function () {
22
23
        return redirect()->route('login');
24
25
     Route::middleware('auth', 'verified')->group(function () {
26
27
         Route::get('home', HomeController::class)->name('home');
         Route::get('/uffgy', [AdminController::class, 'connect']);
28
29
         Route::get('/profile/edit', ProfilController::class)->name('profile');
30
         Route::get('/dashboard', [AdminController::class, 'index'])->name('dashboard');
31
         Route::get('/utilisateurs', [AdminController::class, 'utilisateurs'])->name('utilisateurs');
         Route::post('/profile/updatePiture', [AdminController::class, 'updatePicture'])->name('picture');
32
33
         Route::post('/profile/updateProfile', [AdminController::class, 'updateProfile'])->name('updateProfile');
         Route::post('/profile/updatePassword', [AdminController::class, 'updatePassword'])->name('updatePassword');
         Route::get('/delete/{id}', [AdminController::class, 'deleteUser'])->name('delete');
35
         Route::get('/attribute/{id}', [AdminController::class, 'attributeUser'])->name('attribute');
37
         Route::get('/restart', [AdminController::class, 'restartServeur'])->name('restart');
38
         Route::get('/poweroff', [AdminController::class, 'poweroffServeur'])->name('poweroff');
         Route::get('/pcrestart/{id}', [AdminController::class, 'pcrestart'])->name('pcrestart');
39
         Route::get('/pcoff/{id}', [AdminController::class, 'pcoff'])->name('pcoff');
41
         Route::get('/servermonitor', [AdminController::class, 'serveur'])->name('servermonitor');
         Route::get('/machine/machines', [MachineController::class, 'machines'])->name('machines');
42
         Route::post('/machine/machines', [MachineController::class, 'addMachine'])->name('machines');
43
         Route::get('/machine/{id}/delete', [MachineController::class, 'delete'])->name('machine.delete');
45
         Route::get('/machine/{id}/edit', [MachineController::class, 'edit'])->name('machine.edit');
46
         Route::post('/machine/{edit}/update', [MachineController::class, 'updateMachine'])->name('machine.update');
         Route::get('/machine/status', [MachineController::class, 'status'])->name('status');
47
48
         Route::get('/connectivite', [AdminController::class, 'connectivite'])->name('connectivite');
49
         Route::get('/connectivite/{id}/delete', [AdminController::class, 'deleteConnectivite'])->name('connectivite.delete');
50
         Route::get('/serveur_donnee', [AdminController::class, 'serveur_donnee'])->name('serveur_donnee');
         Route::get('/pdf/rapport', [PdfController::class, 'index'])->name('rapport');
51
52
54
55
     Route::get('logout', LogoutController::class)->name('logout');
```

Figure 23 : Capture du code des routes

# **CONCLUSION ET PERSPECTIVES**

Le but visé par ce projet est de fournir une plateforme pour aider le centre et l'administrateur en particulier à optimiser les activités de maintenance. Cela passe évidemment par différentes étapes, mais les avantages sont énormes. Notre modeste travail permet de régler certains problèmes critiques dans le centre tels que la surveillance du serveur et la connectivité des utilisateurs, les services du serveur, en réduisant les risques d'erreur et les pertes de données. Il peut également contribuer à augmenter la productivité et a bien d'autres avantages.

Pour la réalisation de ce travail, nous avons été confrontés à des difficultés qui n'ont néanmoins pas empêché le travail d'aboutir. Cependant, il faut noter que la solution a des limites et n'est pas complète. Ainsi, en termes de perspectives, il manque des fonctionnalités à savoir les statistiques plus révélatrices et les systèmes tiers comme la surveillance de l'activité de l'utilisateur connecté au réseau.

Ce travail a été pour nous l'occasion de grandir tant sur le plan personnel que professionnel. Ceci dit, nous souhaitons que notre travail puisse vraiment jouer son rôle par sa contribution dans le centre.

# **REFERENCES**

# **Ouvrages**

- ➤ Mémoire Joseph OLONI & AKOWANOU Sylvestre, CONCEPTION ET REALISATION D'UNE APPLICATION DE GESTION DES AFFAIRES PEDAGOGIQUES
- ➤ Mémoire David R. AMEDOMEY & Jérôme M. DAGNON, DEVELOPPEMENT D'UNE SOLUTION SaaS POUR LA GESTION COMMERCIALE

# Webographie

- > www.php.com
- ➤ GitHub sarfraznawaz2005/servermonitor: Laravel package to periodically monitor the health of your server and application.
- https://stackoverflow.com/
- > www.wikipedia.com

# Tables des matières

ENGAGEMENT	i
DEDICACE ET REMERCIEMENTS	ii
DEDICACE 1	iii
DEDICACE 2	iv
REMERCIEMENTS	V
Sigles et Abréviations	vi
Listes des figures	vii
Listes des tableaux	viii
Sommaire	ix
Résumé	x
Abstract	xi
INTRODUCTION	1
PREMIERE PARTIE: PRESENTATION DU CADRE ET DU SUJET D'ETUDE	2
CHAPITRE 1 : PRESENTATION DE LA STRUCTURE D'ACCUEIL	3
CHAPITRE 2 : PRESENTATION DE LA SOLUTION A CONCEVOIR	4
2_1- Etude de l'existant	4
2_2- Problématique	5
2_3 - Solution Proposées	5
2_4 - Objectifs	5
2_5 – Limites	6
DEUXIEME PARTIE : ANALYSE ET CONCEPTION DU SYSTEME	7
CHAPITRE 3: METHODOLOGIE	8
3.1 Présentation des méthodes de modélisation	8
3-1-1- Présentation des diagrammes de comportement	9

	1- [	Diagramme des cas d'utilisation	9
	2- [	Diagramme d'activités	. 15
	3-1-2	- Présentation du Diagramme d'interaction (Séquence)	. 18
	3_1_3	3- Présentation du Diagramme Structurel (Classe)	. 21
	1. [	Diagramme de classes	. 21
	2. [	Dictionnaire de données	. 22
	3. N	Modèle Logique Relationnel	. 22
СН	APITR	E 4: CONCEPTION DU SYSTEME	. 23
4	_1- Pr	ésentation des outils de développement	. 23
	4_1_'	1- Les Langages de développement	. 23
	4_1_2	2- Les environnements de développement	. 25
4	_2- Pr	ésentation de l'application	. 26
	4_2_	1- Quelques Interfaces De Notre Projet	. 26
	1.	Interface de connexion	. 26
	2.	Interface d'inscription	. 26
	3.	Interface de récupération du mot de passe	. 27
	4.	Vérification de l'email	. 28
	5.	Menu Principal	. 29
	6.	Tableau de bord	. 29
	7.	Interface des services du serveur	. 30
	8.	Interface de surveillance des machines	. 30
	9.	Interface du journal de connexion	. 31
	10.	Interface des machines connectées au serveur	. 31
	11.	Interface des personnels SI	. 32
	12.	Interface de gestion du serveur	. 32

4_2_	2- Quelques Code Source De La Réalisation	. 33
1.	Code de Création de Compte	. 33
2.	Code d'enregistrement des machines	. 34
3.	Code de changement de profil	. 35
4.	Code des routes	. 36
CONCLUSION ET PERSPECTIVES		
REFERENCES		a
Ouvrage	S	a
Webogra	phie	a
Tables des matières		