

RAPPORT DE PROJET

SYSTEME DE SUPERVISION RESEAU
ORIENTE CLOUD (A.S.D)

Rédigé et présenté par :

CHARLES PHILIPPE TONYE

Sous l'encadrement

Sous l'encadrement Académique de

Dr Rodrigue DOMGA KOMGUEM

Professionnel de

**Docteur (PhD) en Réseaux et Télécommunications
Informatiques Université de Yaoundé I**

Année 2022- 2023

Sommaire

Sommaire.....	2
Résumé.....	5
Introduction Générale.....	6
INTRODUCTION.....	9
CONTEXTE ET JUSTIFICATION.....	9
CHAPITRE II : CAHIER DES CHARGES.....	9
LES CIBLES ET CONCURRENTS.....	10
OBJECTIFS GENERAUX ET SPECIFIQUES.....	10
Objectif général.....	10
Objectif spécifique.....	10
SPECIFICATION DES BESOINS FONCTIONNELLES.....	11
1-COTE UTILISATEUR.....	11
2-NIVEAU SYSTÈME.....	11
SPECIFICATION DES BESOINS NON FONCTIONNEL.....	12
1. EXIGENCE LIEES A LA PERFORMANCE.....	12
2. EXIGENCES LIEES A LA SECURITE.....	12
Sécurité.....	12
Évolutivité.....	13
Extension de l'application pour prendre en charge de nouvelles fonctionnalités.....	13
Performance.....	13
Capacité à gérer de grandes quantités de données.....	13
Disponibilité.....	13
Mécanismes de redondance pour assurer la disponibilité du système.....	13
PLANIFICATION DU PROJET.....	14
i. Analyse des besoins.....	14
ii. Conception de l'architecture.....	14
iii. Développement des fonctionnalités.....	14
iv. Tests et validation.....	14
v. Déploiement et mise en production.....	14
vi. Livrables.....	16
CONCLUSION.....	16
INTRODUCTION.....	17
I. ETUDE DE L'EXISTANT.....	17
1. La Gestion des machines.....	17
CHAPITRE III : DOSSIER D'ANALYSE.....	17
2. La Gestion des utilisateurs, droits et rôles.....	18

3. La génération des paramètres.....	18
II. CRITIQUES ET LIMITES DE L’EXISTANT	18
III. SOLUTION PROPOSEES	18
IV. CHOIX ET DESCRIPTION DE LA METHODE D’ANALYSE	19
IV. 4.1. Choix	20
IV. 4.2. Présentation du 2TUP	20
V. Diagramme de cas d’utilisation.....	21
CONCLUSION.....	24
INTRODUCTION	25
I. OBJECTIFS DU DOSSIER DE CONCEPTION	25
II. PRESENTATION DES DIAGRAMMES INTERVENANTS	25
CHAPITRE IV: DOSSIER DE CONCEPTION.....	25
II.1. Diagramme de classe	26
II.2. Diagramme d’état de transition.....	27
CONCLUSION.....	27
INTRODUCTION	28
I. CHOIX DES OUTILS	28
CHAPITRE V: MISE EN OEUVRE.....	28
II. DIAGRAMME DE DEPLOIEMENT	29
2.1 L’architecture physique (ou matérielle).....	29
2.2 L’ARCHITECTURE LOGIQUE.....	30
CONCLUSION.....	30
INTRODUCTION	31
CHAPITRE VI: INSTALLATION ETGUIDE D’UTILISATION.....	31
GUIDE D’INSTALLATION.....	31
I. Principe	31
II. Langage.....	31
III. Block.....	31
IV. Fichier.....	32
V. SCHEMA SELON UN POINT DE VUE ABSTRAIT	35
VI. GUIDE D’UTILISATEUR	36
Interfaces de choix de déploiement	36
Déploiement coté Client : Prépare le client à être superviser par le serveur.....	37
Interface de connexion coté serveur.....	37
Interfaces administrateur réseau.....	37
Interface de profil administrateur	38
Interface Ajout d’une machine.....	38

Interface suppression d'une machine	38
Listes des machines d'un réseau	39
Choix de ressource de monitoring.....	39
Tableau de bord de ressources supervisées	40
Quelques cas de Test de monitoring	40
Vérification des tests sur la machine distante	40
CONCLUSION GENERALE.....	41
PERSPECTIVES	41

Résumé

Ce rapport de projet présente les différentes étapes et activités menées pour la mise sur pieds d'une application de supervision réseau orientée Cloud dans une université.

Dans l'optique de faciliter et améliorer le monitoring nous avons envisagé développer une application A.S.D (Application de Surveillance à Distance) dont le but est de permettre un contrôle à distance des machines de l'université de Yaoundé I et en particulier au sein du département d'informatique.

Ceci nécessite le développement d'une application spécifique et adaptée à l'environnement universitaire en corrélation avec les habitudes en présentiel à la fois des enseignants et de l'administration.

Pour atteindre cet objectif nous avons utilisé comme langage d'implémentation le Shell, le C, le Python (Tkinter, Pygame), comme langage de programmation. Il a donc été mis sur pied une solution informatique A.S.D dont les fonctionnalités sont : créer, supprimer, renommer, copier un fichier ou un dossier dans une machine distante, l'éteindre, mettre en veille, redémarrer l'apprenant, sa suppression, affiche l'état de la mémoire, des processus...

Mots-clés : A.S.D, Python, Shell, Cloud.

Introduction Générale

L'évolution de l'informatique a permis l'émergence de nouvelles spécialités, parmi lesquels la gestion des systèmes d'information. En GSI on modélise les informations théoriques en pratique avec rédaction d'un projet de développement logiciel.

Pour mener à bien notre projet, nous réaliserons une série d'étude qui sera recueillie dans ce présent rapport. Lequel sera fait en six grands chapitres successifs dont le cahier des charges, le dossier d'analyse, le dossier de conception, le dossier de réalisation et le guide d'installation et d'utilisation.

Pour amorcer le chapitre 1 cahier des charges illustrera les besoins fonctionnels et non fonctionnels ; le chapitre 2 étant le dossier d'analyse consistera à étudier le problème et à proposer des solutions ; le chapitre 3 qui est le dossier de conception aura pour but de modéliser notre future application et de la structurer ; le chapitre 4 dossier de réalisation nous présentera l'architecture de l'application.



CHAPITRE I : CAHIER DES CHARGES

INTRODUCTION

Le cahier des charges est un document établissant de commun accord entre le maître d'œuvre et le maître d'ouvrage visant à formaliser les besoins liés à la réalisation d'un projet. Il assure l'aspect réglementaire et juridique d'un projet et permet d'organiser le projet en fonction des tâches à exécuter ainsi que la durée du projet. La structure d'un cahier des charges varie selon le cadre du projet ; pour notre projet qui rentre dans la supervision réseau à distance, il sera question dans ce chapitre de présenter tour à tour le contexte, les objectifs généraux et spécifiques, les besoins fonctionnels et non fonctionnels, le modèle de gestion actuels, la planification, le financement, contraintes et les livrables de notre projet.

CONTEXTE ET JUSTIFICATION

Le contexte dans lequel nous réalisons cette application de supervision réseau à distance via le Cloud repose sur deux aspects : Le premier aspect étant de concrétiser par la pratique les différentes notions apprises au différents cours de notre cursus académique et le second aspect étant le constat de faciliter la fluidité des informations entre le département d'informatique et le rectorat. Face à l'évolution rapide des technologies et la mondialisation croissante des entreprises, la supervision des réseaux à distance est devenue essentielle pour assurer le bon fonctionnement et la disponibilité des systèmes. Ce projet intervient dans un **conteste organisationnel**, c'est à dire aide dans un environnement où les machines sont dispersées géographiquement, il est nécessaire de pouvoir superviser et gagner en temps par un contrôle de ces machines à distance de manière efficace et sécurisée.

LES CIBLES ET CONCURRENTS

Si l'on ne connaît pas à tout moment l'état de notre réseau et du serveur alors nous sommes comme un, pilote aveugles. Il existe de nombreux outils-commerciaux pour la surveillance du réseau Parmi les concurrents on a :

- Nagios : écrit en langage C possède une partie serveur très fiable et interface ;
- Grafana : afficher les données issues de promoteurs de
- Prometheus : interfaces web simpliste et pour vérifier le paramétrage du serveur
- Spiceworks Network Monitor idéal pour la surveillance de la mémoire et disque dur, Ntop propose de bon graphique et tableau en temps réel et passée du réseau, NeDi communique via Protocol Telnet : poste les sujets déjà conçu sur sa plateforme.

La différence avec notre logiciel est :

- La mise sur pied d'un dispositif d'installation partielle ou total avant déploiement de l'application,
- Configuration plus facile,
- Nombre minimum de ligne de code,
- Création d'un profil utilisateur dynamique et monitoring en fonction de la date, le jour et l'heure.

OBJECTIFS GENERAUX ET SPECIFIQUES

Objectif général

A la fin de nos travaux il sera question pour nous de concevoir un simulateur A.S.D capable :

- D'assister l'administrateur dans le contrôle des machines,
- De permettre une résolution à temps réels des problèmes des utilisateurs du réseau.

Objectif spécifique

Le but principal de l'application de supervision réseau à distance est de permettre aux administrateurs réseau de surveiller et de gérer les machines à distance de

Manière centralisée, en utilisant les avantages offerts par le Cloud Computing. Cette application fournira des fonctionnalités avancées telles que la collecte en temps réel des données réseau, l'analyse des performances, la visualisation des données, les alertes et notifications, ainsi que la gestion des utilisateurs et des droits d'accès. L'objectif ultime est d'améliorer l'efficacité opérationnelle, d'optimiser la disponibilité des systèmes et de faciliter la prise de décisions basée sur des informations précises et à jour.

SPECIFICATION DES BESOINS FONCTIONNELLES

1-COTE UTILISATEUR

Le client peut :

- Lancer installation totale ou partielle de l'application
- Son adresse @IP.

2-NIVEAU SYSTÈME

Le système doit pouvoir :

- Assurer la mise à jour des données de l'utilisateur
- Enregistrer les comptes des utilisateurs
- Supprimer, renommer, copier, créer un dossier ou fichier, éteindre la machine distante.
- Collecte des données des machines à distance : L'application doit être capable d'établir une connexion sécurisée avec les machines à distance et de collecter en temps réel les données pertinentes, telles que les performances du réseau, l'utilisation des ressources, les statistiques de trafic, etc.
- Collecte et stockage de données réseau
- Collecte de données de trafic réseau : L'application doit être capable de collecter et d'analyser les données de trafic réseau, y compris les paquets réseau, les flux de données, les statistiques de bande passante, les adresses IP source et destination, etc. Cette collecte de données permettra d'obtenir une vue d'ensemble du trafic réseau et de détecter les éventuelles anomalies ou les goulots d'étranglements.
- Analyse et visualisation des données : L'application doit permettre la création de tableaux de bord personnalisables et de graphiques interactifs pour visualiser les données collectées. Ces tableaux de bord doivent fournir une représentation visuelle claire et concise des tendances, des performances et des anomalies du réseau.

- Analyse des tendances et des anomalies : L'application doit être capable d'effectuer des analyses avancées sur les données collectées afin de détecter les tendances, les variations saisonnières, les anomalies de performance, les incidents de sécurité, etc. Ces analyses permettront aux administrateurs réseau d'identifier les problèmes potentiels et de prendre des mesures correctives appropriées.
- Définition des seuils d'alerte : Les administrateurs réseau doivent pouvoir définir des seuils d'alerte personnalisés en fonction des paramètres de performance critiques. Ces seuils peuvent inclure des limites de bande passante, des temps de réponse maximum, des niveaux d'utilisation des ressources, etc.
- Envoi de notifications en cas de dépassement des seuils : L'application doit être en mesure de surveiller en continu les paramètres de performance et d'envoyer des notifications instantanées aux administrateurs réseau lorsque des seuils d'alerte sont dépassés. Les notifications peuvent être envoyées par courrier électronique, SMS ou d'autres moyens de communication appropriés.

SPECIFICATION DES BESOINS NON FONCTIONNEL

1. EXIGENCE LIEES A LA PERFORMANCE

Le système doit être capable sur sa plate-forme de :

- Servir au moins 100 personnes connectées simultanément
- Supporter jusqu'à 20000 requêtes par seconde

2. EXIGENCES LIEES A LA SECURITE

Sécurité

Authentification et autorisation des utilisateurs : L'application doit mettre en place des mécanismes solides d'authentification des utilisateurs, tels que l'utilisation de mots de passe sécurisés, d'authentification à deux facteurs ou d'autres méthodes d'identification robustes. De plus, des mécanismes d'autorisation doivent être mis en place pour garantir que seuls les utilisateurs autorisés ont accès aux fonctionnalités et aux données appropriées.

Chiffrement des données : Les données transitant entre les machines à distance, l'application et le stockage doivent être chiffrées pour garantir leur confidentialité et empêcher tout accès non autorisé. Les algorithmes de chiffrement forts doivent être utilisés pour protéger les données sensibles.

Évolutivité

Capacité à gérer un grand nombre de machines à distance : L'application doit être capable de gérer efficacement un nombre croissant de machines à distance sans compromettre les performances. Elle doit être conçue de manière à pouvoir s'adapter à l'ajout de nouvelles machines et à la croissance du réseau.

Extension de l'application pour prendre en charge de nouvelles fonctionnalités L'architecture de l'application doit être modulaire et extensible afin de permettre l'ajout de nouvelles fonctionnalités et de modules complémentaires à mesure que les besoins évoluent. Cela garantira la flexibilité et la pérennité de l'application à long terme.

Performance

Temps de réponse rapide lors de l'accès aux données : L'application doit fournir des temps de réponse rapides lors de l'accès aux données, que ce soit pour afficher des indicateurs de performance, générer des rapports ou effectuer des analyses. Les requêtes et les opérations doivent être optimisées pour minimiser les délais de traitement.

Capacité à gérer de grandes quantités de données

L'application doit être capable de gérer efficacement de grandes quantités de données collectées, notamment en termes de stockage, de traitement et de recherche. Des mécanismes d'optimisation des performances, tels que l'indexation des données et la mise en cache, doivent être mis en place pour garantir des performances optimales, même avec des volumes importants de données.

Disponibilité

Surveillance et détection des pannes : L'application doit être équipée de mécanismes de surveillance et de détection des pannes pour détecter rapidement les éventuelles défaillances du système. Des alertes doivent être générées pour informer les administrateurs réseau en cas de panne ou de dysfonctionnement.

Mécanismes de redondance pour assurer la disponibilité du système

L'application doit être conçue pour être hautement disponible, en utilisant des mécanismes de redondance tels que la réplication des serveurs, la mise en cluster et la répartition de charge. Cela garantira que l'application reste accessible même en cas de défaillance d'une partie du système.

PLANIFICATION DU PROJET

La planification d'un projet est très importante lors de sa réalisation car elle consiste à déterminer et ordonnancer les tâches du projet. Ce qui permettra de déterminer si les objectifs sont réalisés et entre autres suivre et communiquer l'avancement du projet. Nous avons ainsi ressorti la planification de l'évolution de notre projet.

i. Analyse des besoins

Dans cette phase, les besoins et les exigences fonctionnelles et non fonctionnelles de l'application seront identifiés et documentés. Une compréhension claire des objectifs et des cas d'utilisation sera établie.

ii. Conception de l'architecture

Une fois les besoins identifiés, l'architecture globale de l'application sera conçue. Cela comprendra la définition des composants, des flux de données, des interactions et des interfaces.

iii. Développement des fonctionnalités :

Les fonctionnalités de l'application seront développées en suivant les bonnes pratiques de développement logiciel.

Nous travaillerons sur les différentes parties de l'application, en veillant à ce qu'elles s'intègrent correctement.

iv. Tests et validation :

Des tests approfondis seront effectués pour s'assurer que l'application fonctionne conformément aux exigences spécifiées. Cela inclura des tests unitaires, des tests d'intégration et des tests de performance.

v. Déploiement et mise en production

Une fois les tests terminés, l'application sera déployée sur l'infrastructure Cloud choisie. Des procédures de déploiement et de mise en production efficaces seront suivies pour garantir un déploiement sans heurts.

Il se résume par le diagramme de Gantt qui présente les tâches ou activités à réaliser et les temps nécessaires pour leur réalisation.

Diagramme de Gants

Numéro	Tâche	Date début	Date de fin	Durée								
1	Rédaction du rapport	01/04/2023	06/06/2023	66								
2	Dossier d'insertion	05/04/2023	12/04/2023	07								
3	Cahier de charge	14/04/2023	28/04/2023	14								
4	Cahier d'analyse	02/05/2023	09/05/2023	07								
5	Cahier de conception	11/05/2023	18/05/2023	07								
6	Mise en œuvre	23/05/2023	06/06/2023	38								

Financement/contraintes
Tableau : coût des ressources

RESSOURCES	NOMS DES RESSOURCES	CARACTERISTIQUES	PRIX UNITAIRE	PRIX TOTAL
Matérielles	02 ordinateurs portables	HP Corei5 Disque dur 500giga Ram 4giga Processeur 2.30GHZ	300 000FCFA	900 000FCFA
	01modem	Camtel 4G	20 000FCFA	20 000FCFA
	01 Rame de Papier	Format A4 double A	3000FCFA	3000FCFA
	04Stylos	Schneider	200FCFA	800FCFA
Logiciels	Tkinter	Fenêtres, widgets	Gratuit	/
	Pygame	Gestion Image	Gratuit	/
	MySQL	Version 8.0.16	Gratuit	/
	Adobe Acrobat Reader DC	Version 21.005.20048	Gratuit	/
	Suite Microsoft Office	2016	20 000FCFA	20 000FCFA
	Antivirus	Kaspersky	30 000FCFA	30 000FCFA

vi. Livrables

En gestion de projet, un livrable désigne tout produit fourni pendant la réalisation du projet et nécessaire pour atteindre les objectifs. C'est aussi l'ensemble des composants matérialisant le résultat d'une prestation de réalisation : **c'est le résultat tangible d'une production réelle, appréhendable et mesurable**. Les livrables de notre projet sont : une application fonctionnelle, un manuel utilisateur et guide d'installation ; le rapport du projet.

CONCLUSION

Parvenu au terme de ce chapitre qui consistait à présenter les éléments qui vont être déployés dans la mise en production de notre application. Dans le prochain chapitre il sera question pour nous de faire une analyse complète du système A.S.D.



CHAPITRE II : DOSSIER D'ANALYSE

INTRODUCTION

En génie logiciel la phase d'analyse est celle qui précède la phase de conception et qui suit le cahier de charges. Cette phase doit être abordée avec une attention particulière car c'est elle qui détermine la faisabilité du projet. Le but de ce projet étant de mettre sur pied une application de supervision réseau à distance orienter Cloud dans une université, nous ne saurions concevoir cela sans au préalable étudier l'existant afin d'en déceler les limites et de proposer les solutions pour une application adéquate et performante. Ainsi donc ce dossier d'analyse consistera tour à tour à faire un choix sur la méthode d'analyse qui nous aidera à mener à bien notre projet mener une étude de l'existant, présenter les critiques et les limites de l'existant, ressortir la problématique et tout en ressortissant les diagrammes intervenants.

I. ETUDE DE L'EXISTANT

Dans notre travail, on s'intéresse aux **services administrateur/client**, qui sont à la base de la réalisation de ce projet, ces services sont dirigés par un administrateur qui aura comme rôle de gérer les différentes privilèges et accès de l'application, cette gestion traite les informations personnelles et professionnelles des utilisateurs, et les affaires qui sont en relations avec ces derniers. Énumérons quelques taches réalisées dans ces services :

1. La Gestion des machines

Ce module doit nous permettre de :

- Gérer la création et suppression des machines du réseau à monitorer
- Enregistrer le nom d'une machine et son numéro de port, @IP.

2. La Gestion des utilisateurs, droits et rôles.

Ce module doit nous permettre de :

- Gérer les différents utilisateurs de notre système (création de compte, modification, suppression, attribution des droits, retrait des droits etc. ...)
- Gérer les administrateurs (administration, enseignant)

3. La génération des paramètres.

Ce module doit nous permettre de :

- Changer de profil, de choisir une machines précise à superviseur,
- Initialiser son mot de passe après chaque nouvelle connexion.

II. CRITIQUES ET LIMITES DE L'EXISTANT

Pour arriver à mettre en place un système A.S.D, il est indispensable de faire une étude complète pour les processus de travail existants dans l'entreprise.

Cette étude doit détecter les insuffisances et les difficultés qui peuvent mener à des défaillances dans le processus du travail et toucher à la qualité de la gestion des ressources d'informations. Alors la réalisation du projet se basera sur les défaillances qui existent dans la méthode de travail suivie.

Parmi les difficultés détectées dans l'existant, on peut citer :

- Risque de l'indisponibilité des informations pendant les congés ; le service est fermé, donc l'accès aux informations par les employés sera limité.
- La charge du travail devienne très importante vue l'absence d'un vrai système qui centralise les informations.

III. SOLUTION PROPOSEES :

❖ Besoins et objectifs :

- Essayer de centraliser les informations sous un seul système.
- Faciliter l'accès aux documents mis à la disposition des utilisateurs.
- Assurer un niveau de sécurité lors du traitement des informations.

❖ Solution envisagée :

Pour atteindre les objectifs mentionnés précédemment, on est mené à réaliser une application qui peut réaliser ces objectifs, et ceci en offrant :

- Une base de données centralisée, dans laquelle on pourra sauvegarder et traiter les informations.
- Une base d'informations accessible à tous les futurs utilisateurs du système, en garantissant un niveau de sécurité et confidentialité pour les informations personnelles.

IV. CHOIX ET DESCRIPTION DE LA METHODE D'ANALYSE

En informatique, une méthode d'analyse est un procédé qui a pour objectif de Formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client.

Le choix de la méthode nécessite une attention particulière car c'est cette méthode choisie qui nous guidera tout au long de la suite de notre travail afin d'obtenir une application fonctionnelle respectant les objectifs visés et pouvant satisfaire aux besoins de l'utilisateur.

Les années 1970 laissent voir les premières méthodes d'analyse. On compte de nos jours plusieurs méthodes parmi lesquels Booch, merise, OOSE, ... et le langage de modélisation UML. Cependant les plus utilisés sont la méthode d'analyse merise et le langage de modélisation UML ; Le tableau ci-dessous présente une analyse comparative de ces deux méthodes.

Tableau : comparaison entre MERISE et UML.

<u>MERISE:</u>	<u>UML:</u>
Méthode d'étude et de réalisation informatique pour les systèmes d'entreprises	Unified Modeling Language
MERISE est une méthode systémique d'analyse et de conception de systèmes d'information. C'est à dire qu'elle utilise une approche systémique	UML n'est cependant pas une méthode mais plutôt un langage de modélisation objet auquel il faut associer une démarche pour en faire une méthode.
MERISE propose de considérer le système réel selon deux points de vue: <ul style="list-style-type: none"> ➤ Une vue statique (données) ➤ Une vue dynamique (traitements) C'est-à-dire qu'avec la méthode MERISE, nous avons une étude séparée des données et des traitements ;	UML propose une approche différente de celle de MERISE en ce sens qu'il associe les données et les traitements. Car avec UML, centraliser les données d'un type et les traitements associés permettent de limiter les points de maintenance dans le code et facilite l'accès à l'information en cas d'évolution du logiciel. De plus, UML décrit la dynamique du système d'information comme un ensemble d'opérations attachées aux objets du système.

IV. 4.1. Choix

Ainsi pour mener à bien notre travail, nous avons choisi le langage de modélisation unifié UML qu'il est un support de communication, un cadre méthodologique (un gain de précision, un gage de stabilité, encourage l'utilisation des outils, facilite la compréhension). Cependant UML n'étant pas une méthode, il est nécessaire d'associer à ce langage de modélisation une méthode. Notre choix s'est porté sur le 2TUP.

IV. 4.2. Présentation du 2TUP

La modélisation UML se base généralement sur les processus unifiés. Dans notre travail on a choisi de suivre la méthode 2TUP (2 Trucks Unified Process) qui se base sur le modèle en Y.

Cette méthode doit suivre et rendre des réponses sur les différents changements subis sur le système d'information en se base principalement sur deux contraintes ; sur le chemin technique et le chemin fonctionnel.

La méthode 2TUP, commence tout d'abord par la préparation d'une étude préliminaire dans laquelle on extrait les acteurs qui vont réagir avec le système à mettre en place, cette réaction se résume dans des interactions Acteurs-Système sous forme d'opérations et d'échanges d'informations afin de satisfaire les besoins et atteindre les objectifs souhaités. Le reste du travail sera décomposé sur trois phases :

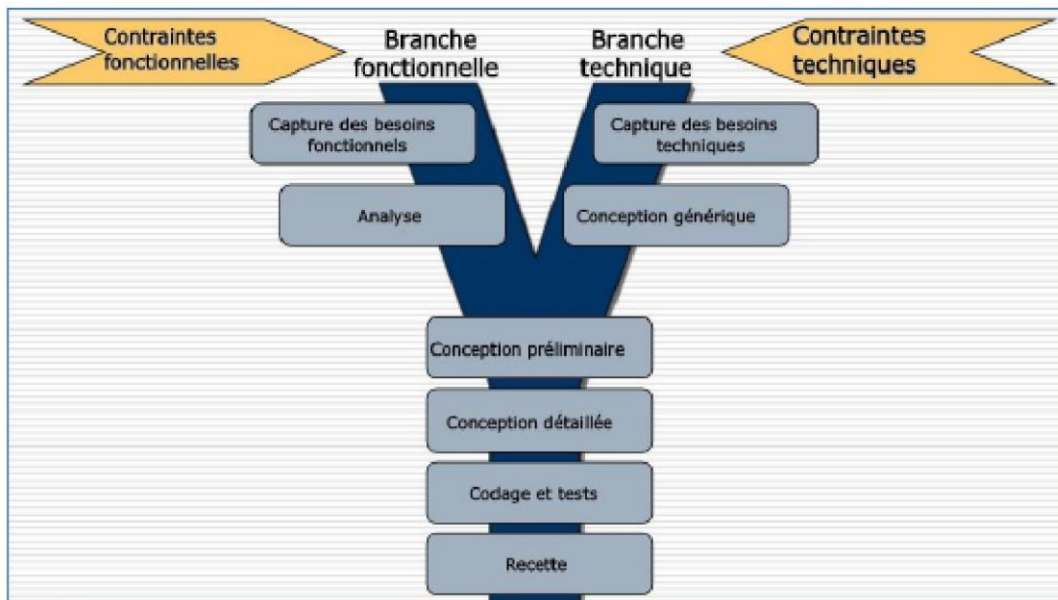
- **La première phase** ; une combinaison entre la branche fonctionnelle et la branche technique ; Où on commence, principalement, par l'extraction et l'étude préliminaire des besoins fonctionnels et des besoins techniques ; là où on fait une étude et analyse générale des besoins.

- **La deuxième phase** ; une combinaison entre la branche fonctionnelle et la branche technique ; et là où on passe à l'analyse et la conception générale du système.

- **La troisième phase** ; la réalisation

À ce niveau on se concentre sur la conception détaillée du système, ensuite le codage et la maintenance du système.

On présente, dans la figure ci-dessous, la méthode de modélisation en Y :



(Diagrammes de la branche fonctionnelle)

Figure : modèle en y

V. Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisations identifient les fonctionnalités fournies par le système (cas d'utilisation), les utilisateurs qui agissent avec le système (acteurs) et les interactions entre ces derniers. Les cas d'utilisation sont utilisés dans la phase d'analyse pour définir les besoins de haut niveau du système. Les objectifs principaux de ce diagramme sont :

- Fournir une vue de haut niveau de ce que fait le système
- Identifier les utilisateurs du système
- Déterminer L'IHM

Figure : description textuelle du cas d'utilisation



VI Diagrammes de séquence

Les diagrammes de séquence documentent les interactions à mettre en œuvre entre les classes pour réaliser un résultat, tel qu'un cas d'utilisation. UML étant conçu pour la programmation orientée objet, ces communications entre les classes sont reconnues comme des messages. Le diagramme de séquence énumère des objets horizontalement, et le temps verticalement. Il modélise l'exécution des différents messages en fonction du temps.

Représentation : dans un diagramme de séquence, les classes et les acteurs sont énumérés en colonne avec leurs lignes de vie verticales indiquant la durée de vie de l'objet.

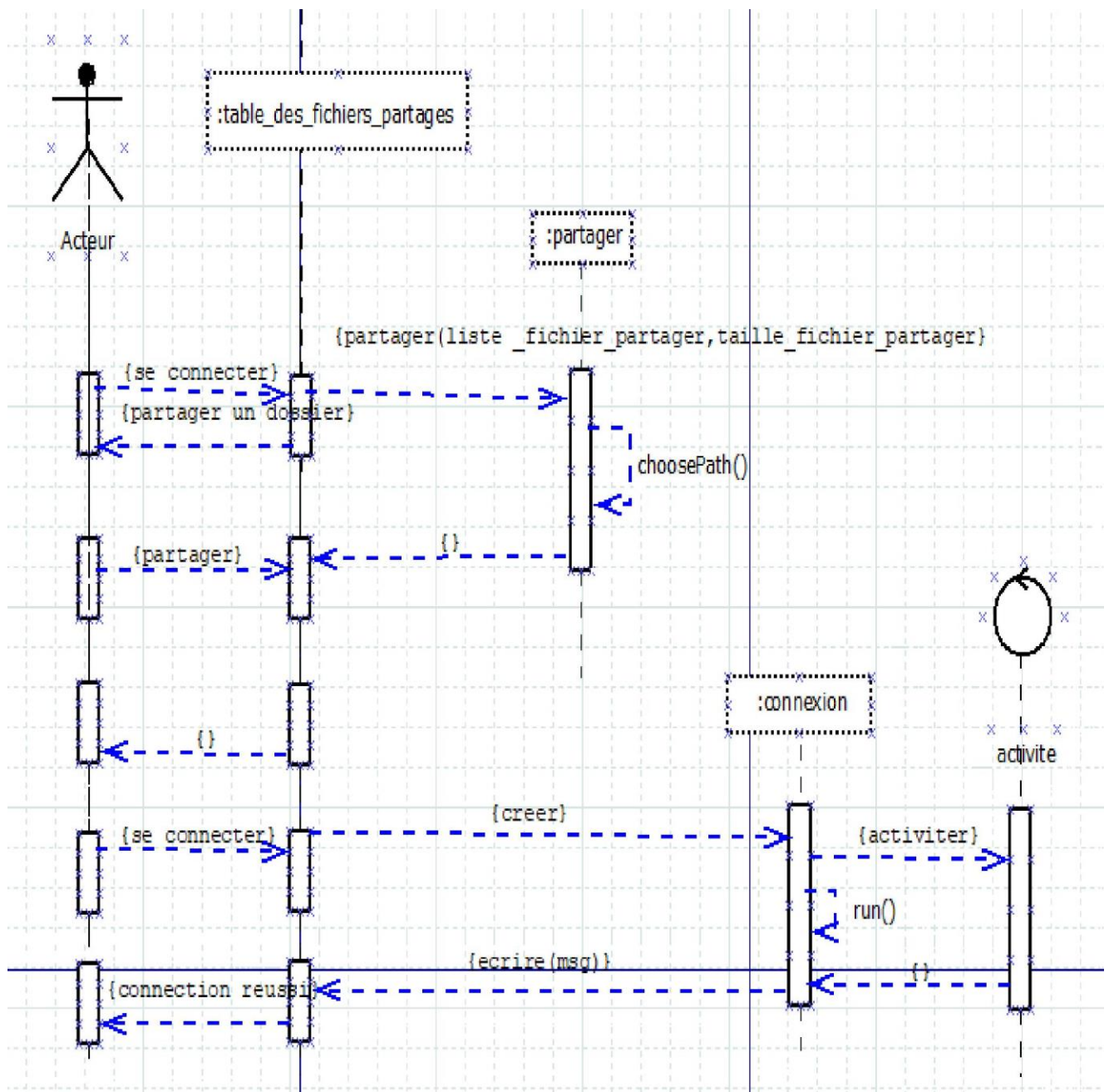


Figure : diagramme séquence du cas d'utilisation connexion

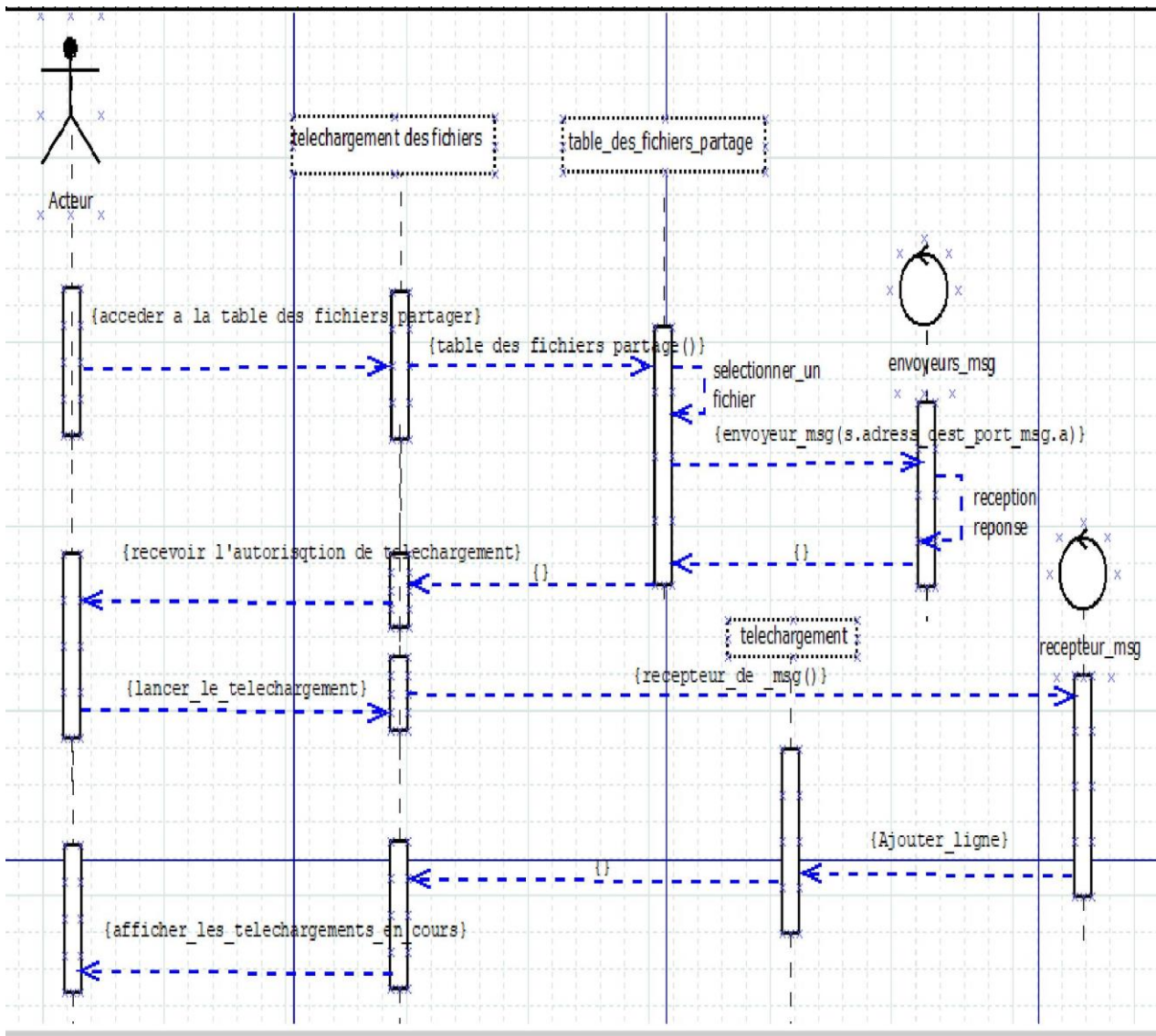


Figure : diagramme séquence du cas d'utilisation téléchargement des fichiers

CONCLUSION

Parvenu au terme de cette partie qui avait pour but de mener les études sur l'existant, ressortir ses limites en proposant des solutions et établir les diagrammes de cas d'utilisation, séquence et classe. Il sera question pour nous par la suite d'établir le dossier de conception qui portera sur l'objectif du dossier de conception ainsi que la présentation du diagramme de classe et d'état de transition.



INTRODUCTION

En gestion de projet, le dossier de conception constitue une étape incontournable. Il nous permet de visualiser et décrire les exigences du système. Il sera question pour nous dans cette partie de présenter l'objectif du dossier de conception ainsi que le diagramme de classe et d'état de transition.

I. OBJECTIFS DU DOSSIER DE CONCEPTION

Le dossier de conception permet de présenter la structure du futur système. Il permet de : Dégager les grandes fonctionnalités du système, Présenter les rapports entre les classes afin de visualiser le futur système, Spécifier les détails de chaque module

II. PRESENTATION DES DIAGRAMMES INTERVENANTS

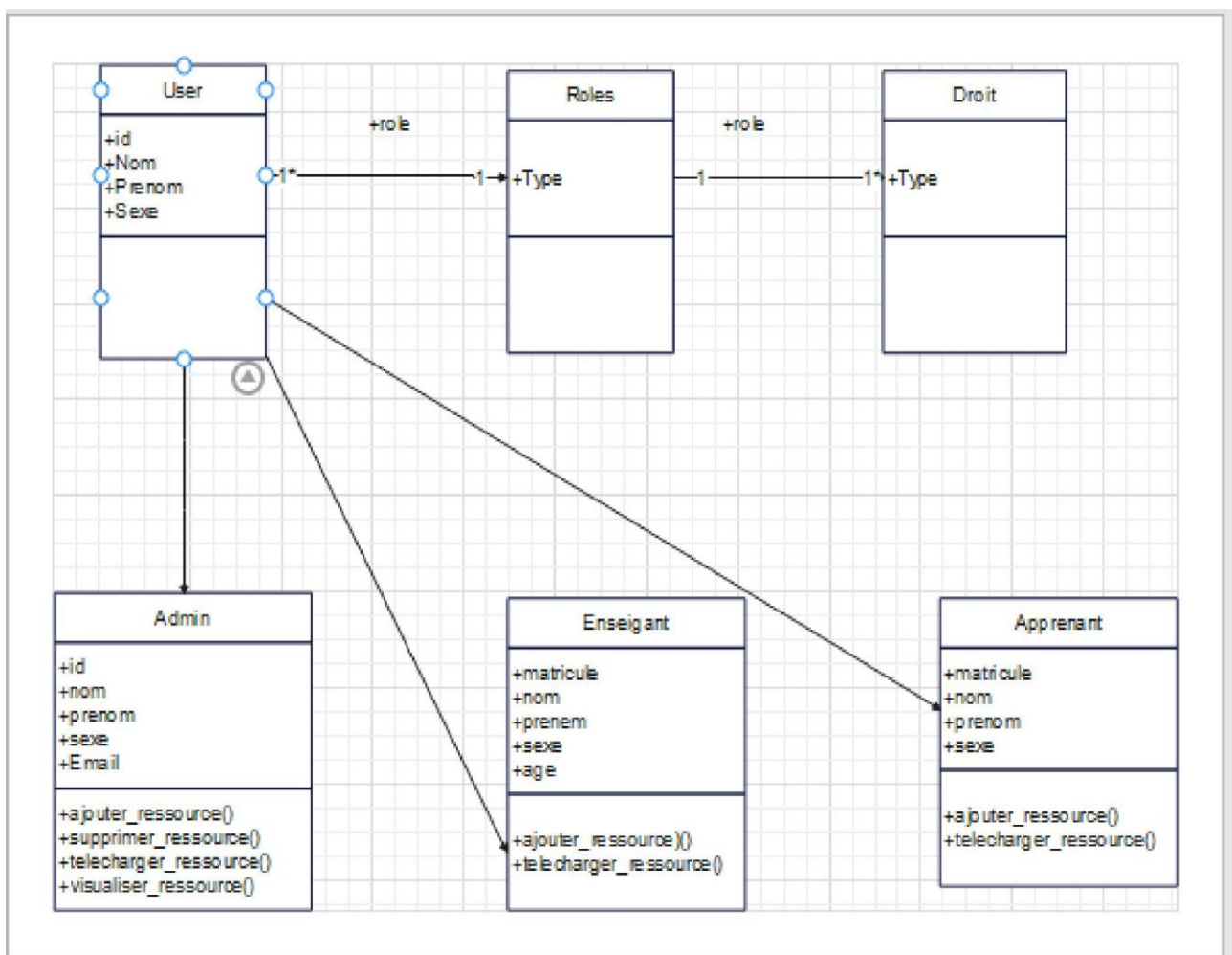
Les diagrammes intervenants dans cette partie sont le diagramme de classe et le diagramme d'état de transition.

II.1. Diagramme de classe

II.1.1. Définition

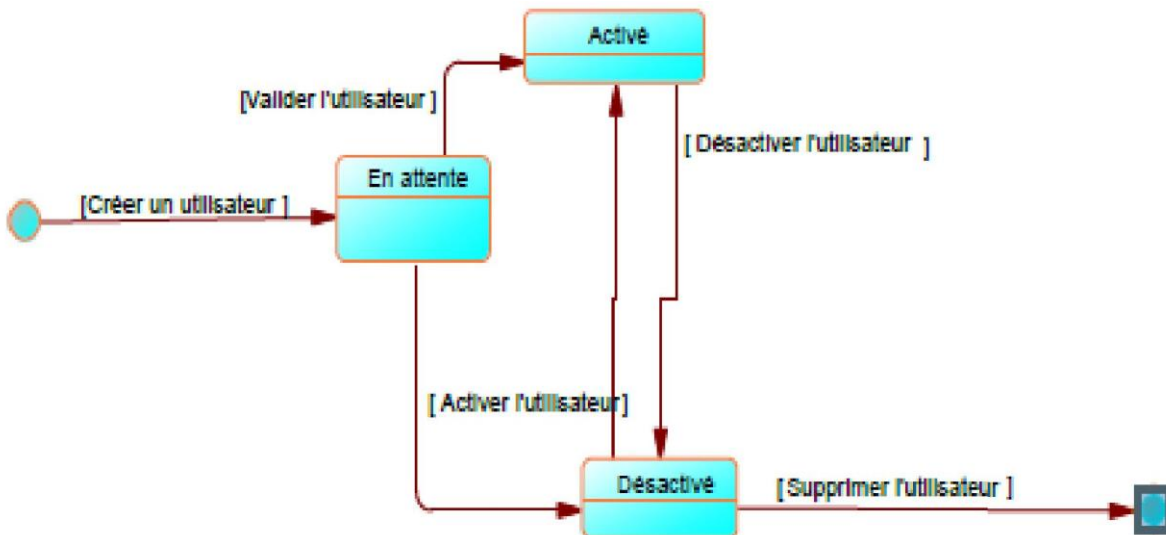
Le diagramme de classe exprime la structure statique du système en termes de classe et de relations entre ces classes. L'intérêt du diagramme de classe est de Modéliser les entités du système d'informations. Le diagramme de classe permet de présenter l'ensemble des informations finalisées qui sont gérés par le domaine. Ces informations sont structurées, c'est à-dire qu'elles sont regroupées dans les classes. Le diagramme de classe comporte quelques concepts : classe, attribut, identifiant, opération (méthode) et relation.

II.1.2. Gestion des utilisateurs dans le réseau UYDEI



II.2. Diagramme d'état de transition

Les diagrammes d'états sont utilisés pour documenter les divers modes qu'une classe peut prendre, et les évènements qui causent une transition d'état. Ils ont pour rôle de représenter les traitements (opérations) qui vont gérer le domaine étudié. Ils Définissent l'enchaînement des états de classe et font Donc apparaitre l'ordonnancement des travaux.



CONCLUSION

Parvenu au terme de cette partie qui avait pour but de présenter l'objectif du dossier de conception ainsi que le diagramme de classe et d'état de transition. il sera question dans la prochaine partie à savoir le dossier de réalisation de Présenter les technologies utilisées ainsi que le diagramme de déploiement.



CHAPITRE IV : MISE EN OEUVRE

INTRODUCTION

La phase de mise en œuvre renvoie au dossier de réalisation de la solution. Il permet de décrire de manière physique l'application c'est l'aboutissement de tous les dossiers précédant. Notre travail dans ce dossier consistera à présenter les technologies utilisées ainsi que le diagramme de déploiement.

I. CHOIX DES OUTILS

Le choix des outils nécessite une attention particulière car nous permettra de visualiser notre application sur le plan structurel et aussi environnemental. Nous avons recueilli l'ensemble d'outils nécessaires à la réalisation de notre application dans le tableau ci-après.

1.1. Environnement matériel

Matériel	Quantité
Ordinateur HP	02
Disque Dur 500 giga	
Ram 4giga	
Processeur 2.6 GHZ	
Modem	01

1.2. Environnement logiciel

Types	noms
Systèmes d'exploitation	Windows10, Linux 18.04
Système de gestion de base de données	MySQL
Logiciel de modélisation	StarUML, Power AMC
Environnement de développement	Gedit, Vim
Logiciel de traitement de texte	Microsoft office Word 2010, 2013

1.3. Langage de programmation

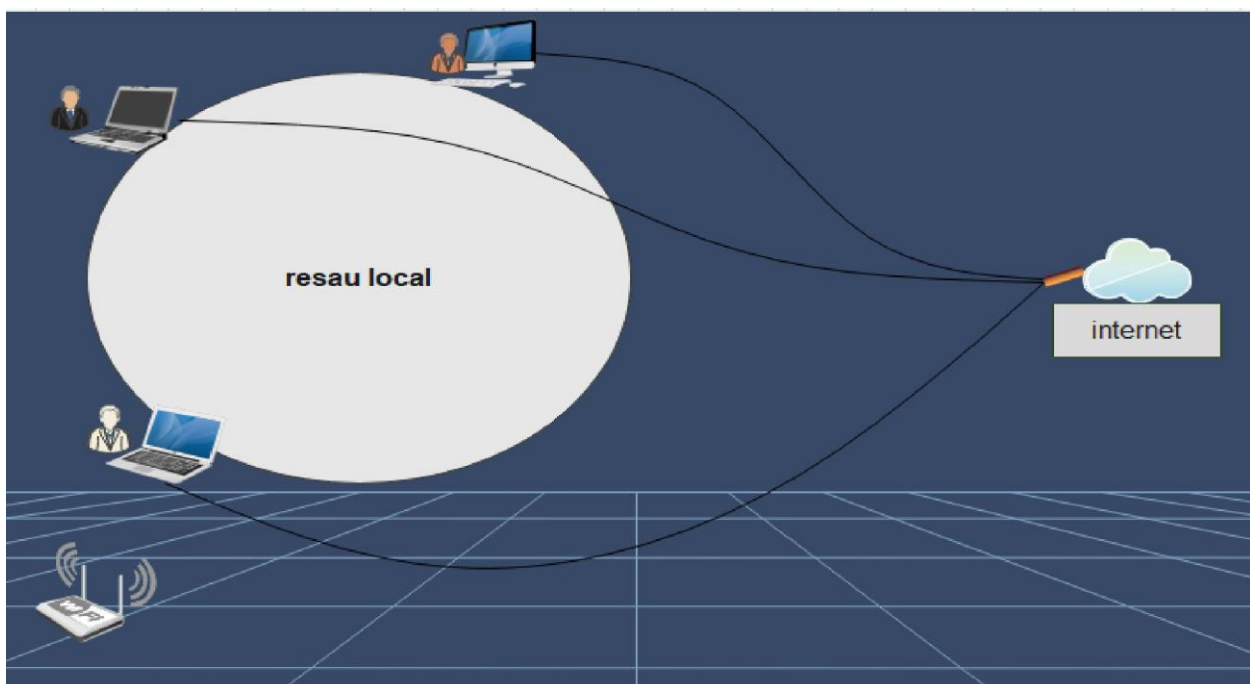
Nous avons choisi de développer notre application avec le Bash, langage C, Python, Tkinter

II. DIAGRAMME DE DEPLOIEMENT

Le diagramme de déploiement modélise les composants matériels utilisés pour Implémenter un système et l'association entre ces composants. Des diagrammes de Déploiement peuvent être mise en œuvre dès la phase de conception pour documenter l'architecture physique et logique du système.

2.1 L'architecture physique (ou matérielle)

Elle décrit les composants concrets : les ordinateurs (postes clients), les téléphones mobiles, le stockage, le serveur local qui communique avec la base de données de l'application, etc.



2.2 L'ARCHITECTURE LOGIQUE

Pour le cas d'espèce nous allons opter pour une architecture 3-tiers ou a 3 niveaux car permet : une plus grande flexibilité/souplesse. Une grande sécurité pour chaque service et de meilleures performances (les taches sont partagées).

CONCLUSION

Parvenu au terme de cette partie où il était question pour nous de présenter les technologies utilisées. Il y en ressort que nous développerons notre application sur Python3-Tk qui sera relié au bash et langage C. Ainsi, la phase d'implémentation de notre application fera l'objet de la prochaine partie.



CHAPITRE V: INSTALLATION ET GUIDE D'UTILISATION

INTRODUCTION

Le guide d'installation et guide d'utilisateur sont des documents qui sont rédigés après la réalisation d'une application. Ils permettent respectivement de montrer le procédé d'installation de l'application et les informations sur son usage. Cette partie consistera à présenter tour à tour le guide d'installation et le guide d'utilisateur.

GUIDE D'INSTALLATION

I. Principe

Notre application permet de surveiller notre machine, une machine ou plusieurs machines à distance en donnant des informations précise récupérer par des programmes en bash (dossier Shell), afficher par des programmes en python (dossier Python) et stocké dans le dossier Data

II. Langage

Nous avons

- Le bash dans le dossier **Shell/**
- Le python dans le dossier **Python/**

III. Block

On peut diviser selon le type de fenêtre python :

- Pygame (import pygame) : Premier.py, lancement_admin.py, lancement_client.py

- Tkinter (import tkinter) : Tous les autres.

IV. Fichiers

a. ASD_installation.exe

Ce fichier permet d'avoir le menu pour installer les différents paquets nécessaires au fonctionnement de l'application et de lancer le programme

b. ASD_executable.exe

Ce fichier est un intermédiaire pour permettre de lancer la première interface graphique tout en cachant les autres parties du programme si on vient à ouvrir le dossier.

c. Premier.py

Ce fichier est l'interface graphique de base qui sera ouvert temps que l'application est active et permet ainsi de donner l'illusion que le programme se fait sur une seule fenêtre. Il suffit de cliquer dessus pour commencer à utiliser l'application proprement dit.

d. mode_de_lancement.py

Ce fichier permet de choisir quel sera l'usage de l'application pour cette fois (Surveillant ou surveillé).

e. lancement_admin.py

Ce fichier lance le programme en tant qu'administrateur (Surveillant) mais il permet clairement de démarrer le ssh qui permet le transfert et la communication entre les machines.

f. lancement_client.py

Ce fichier lance le programme en tant que client (Surveillé) mais il permet clairement de démarrer le ssh qui permet le transfert et la communication entre les machines.

g. info.py

Ce fichier est l'interface qui donne l'aperçu des informations que l'administrateur aura besoin pour connecter notre client à sa liste de machine

h. connexion.py

Ce fichier est l'interface ou on s'identifie pour accéder à notre espace de travail (avec un nom de compte et un mot de passe)

i. oublier.py

Ce fichier est l'interface avec laquelle on peut récupérer un compte donc on ne connaît plus le mot de passe grâce à l'indice donné lors de la création de notre compte administrateur.

j. creer_un_compte.py

Ce fichier est l'interface avec laquelle on crée notre compte administrateur en rentrant les infos nécessaires.

k. option.py (inutilisé)

Ce fichier est l'interface où on a choisi les informations qu'on veut avoir

l. option_alternative.py (inutilisé)

Ce fichier est l'interface qui montre les options qui sont en cours et ce qui ne le sont pas.

m. menu_principale.py

Ce fichier est l'interface principale où on peut lancer la partie de surveillance et faire tous les réglages nécessaires ; C'est l'espace de travail de l'administrateur.

n. menu_machine.py

Ce fichier est l'interface qui permet de voir toutes les machines (clients) qui sont connectées à notre administrateur donc tout ceux qu'il peut surveiller

o. nouvelle_machine.py

Ce fichier est l'interface qui permet d'ajouter une machine avec les informations données par le fichier info.py

p. supprime_machine.py

Ce fichier est l'interface qui permet de supprimer une machine de la liste grâce à son nom.

q. fenetre_resources.py et fenetre_resources (copy).py(inutilisé)

Ces des anciens systèmes d'affichage des informations récupérer par le programme

r. Fenetre_ressources.py

Ce fichier gérer l'affichage des différentes fenêtres qui vont afficher les informations récolter.

s. Connexion.py

Ce fichier est l'interface qui donne les différents temps de connexion.

t. Infos.py

Ce fichier est l'interface qui donne l'information liée à la machine surveillée.

u. Processus.py

Ce fichier est l'interface qui affiche les processus.

v. Manipulation.py

Ce fichier est l'interface qui permet de manipuler la machine en local.

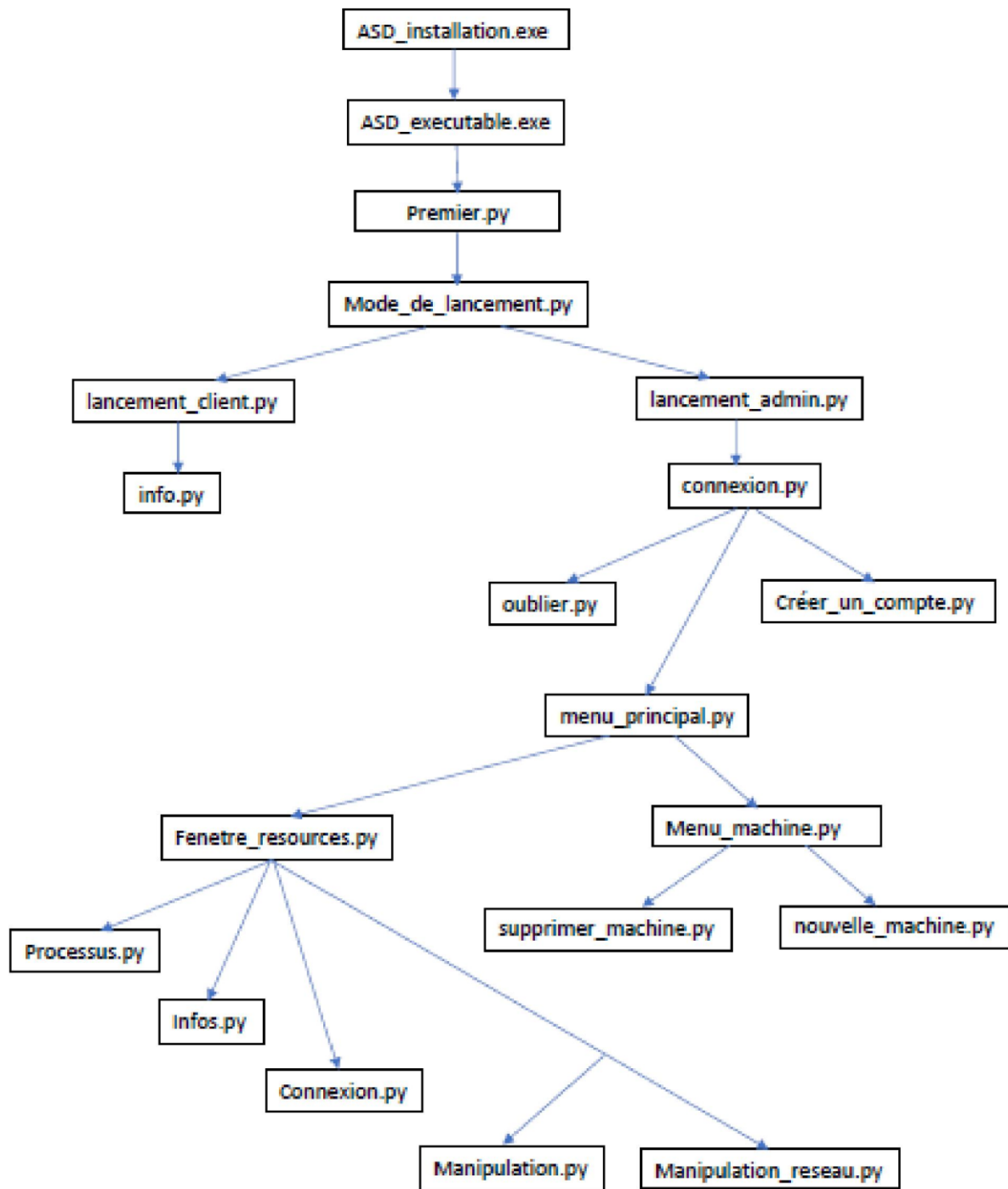
w. Manipulation réseau

Ce fichier est l'interface qui permet de manipuler la machine en réseau.

x. Decision.py (inutilisé)

Ce fichier est le fichier qui gère les choix d'option et option_alternative.

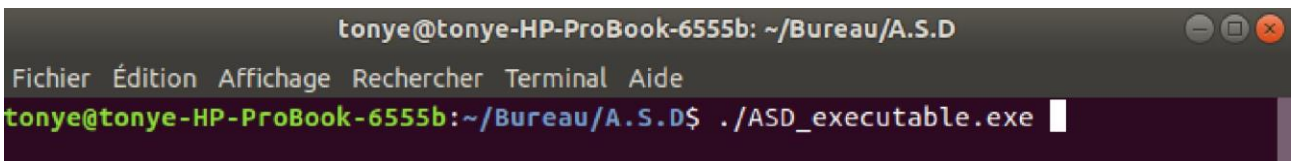
V. SCHEMA SELON UN POINT DE VUE ABSTRAIT



VI. GUIDE D'UTILISATEUR

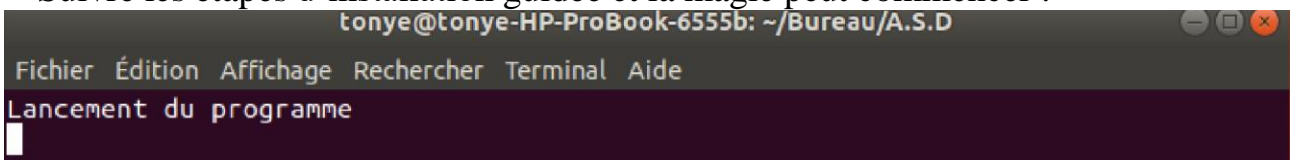
Le guide d'utilisateur explique la démarche d'utilisation de notre application en local via un accès point par ssh à travers les sockets. Dans un premier temps rassurer vous que le dossier A.S.D soit sur le bureau de votre environnement linux. Dans un second temps remplacer les adresses @IP_client puis @IP_serveur respectivement dans le chemin :

/home/login_user/Bureau/A.S.D/ASD_installation/ASD_executable/Python/
lancement_Admin.py et lancement_client.py. Ensuite lancer le fichier
ASD_executable.exe



```
tonye@tonye-HP-ProBook-6555b: ~/Bureau/A.S.D
Fichier Édition Affichage Rechercher Terminal Aide
tonye@tonye-HP-ProBook-6555b:~/Bureau/A.S.D$ ./ASD_executable.exe
```

-- Suivre les étapes d'installation guidée et la magie peut commencer :

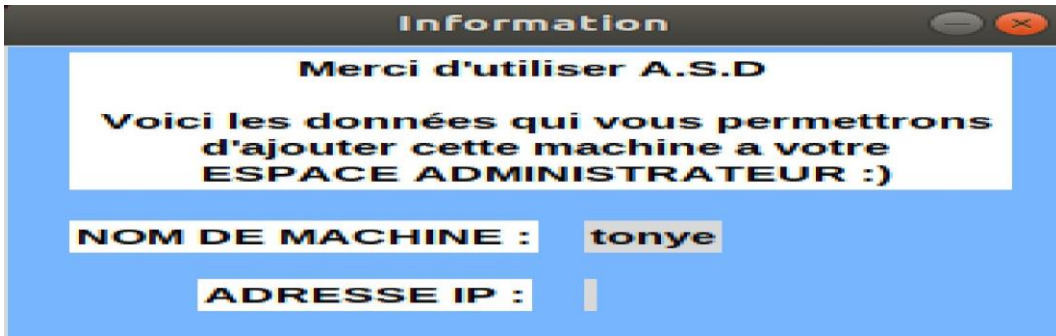


```
tonye@tonye-HP-ProBook-6555b: ~/Bureau/A.S.D
Fichier Édition Affichage Rechercher Terminal Aide
Lancement du programme
```

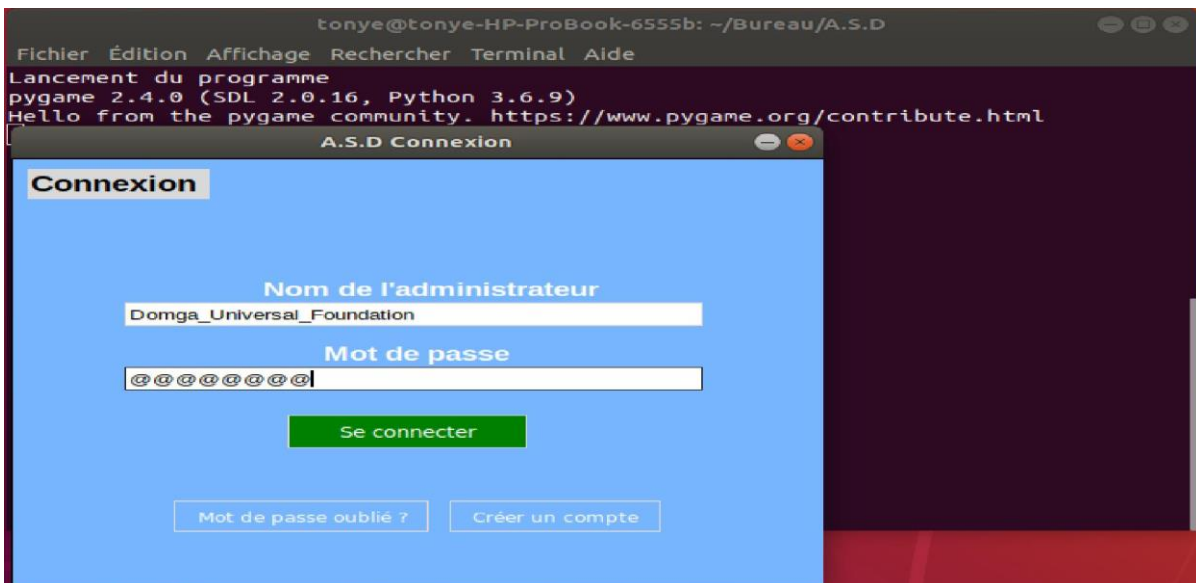
Interfaces de choix de déploiement :



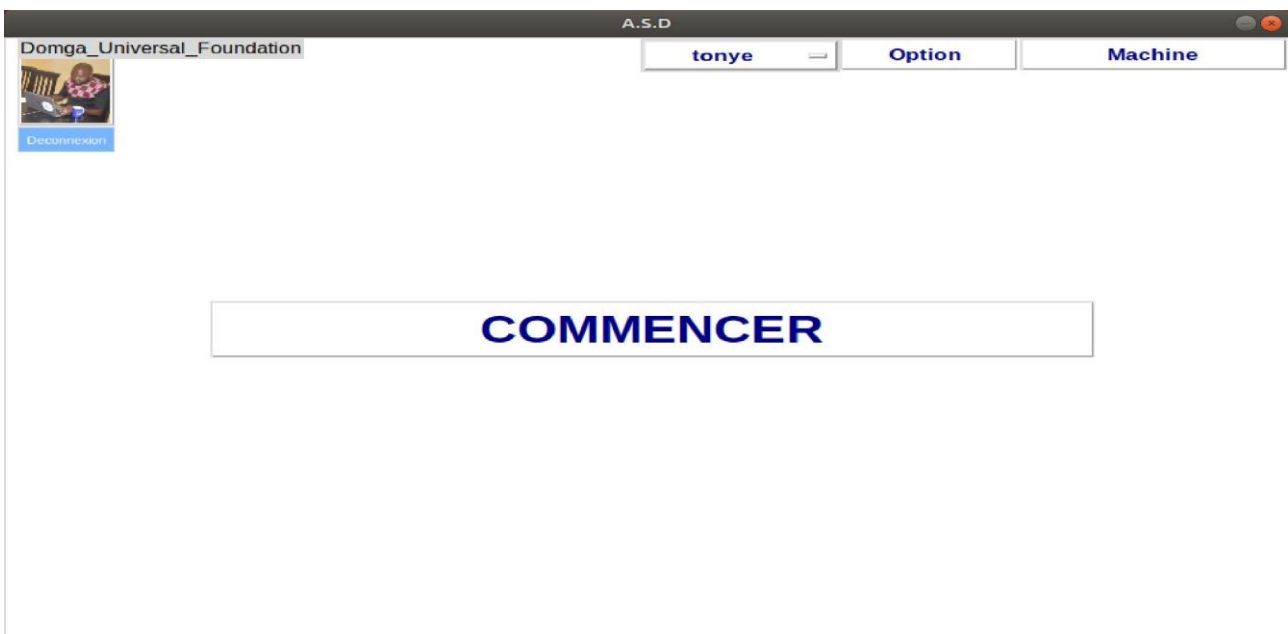
Déploiement coté Client : Prepare le client à etre superviser par le serveur



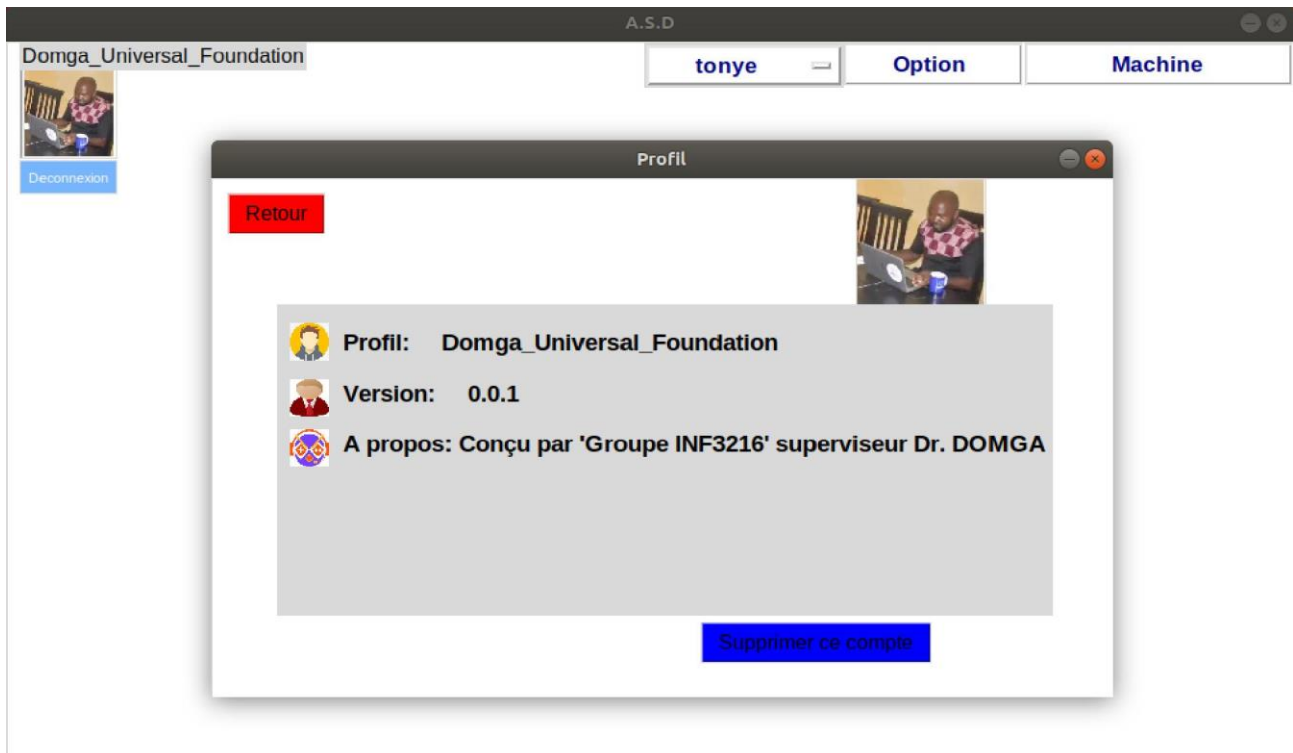
Interface de connexion coté serveur



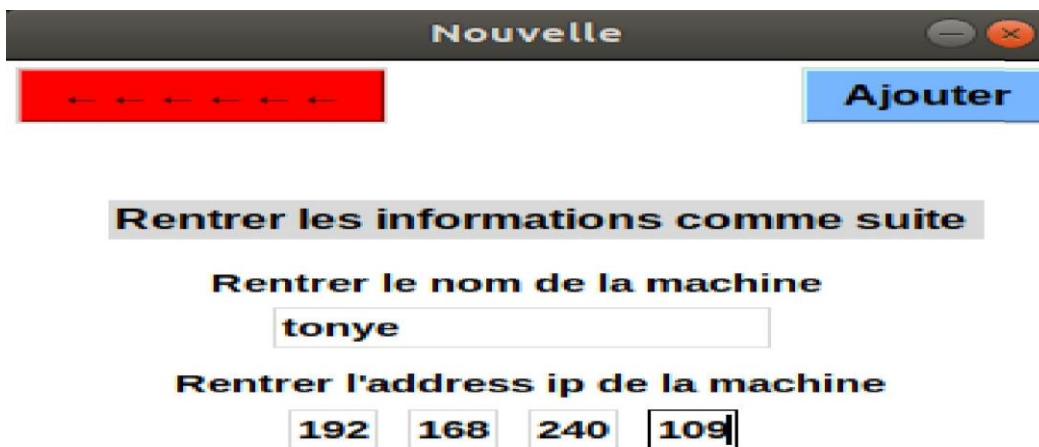
Interfaces administrateur réseau :



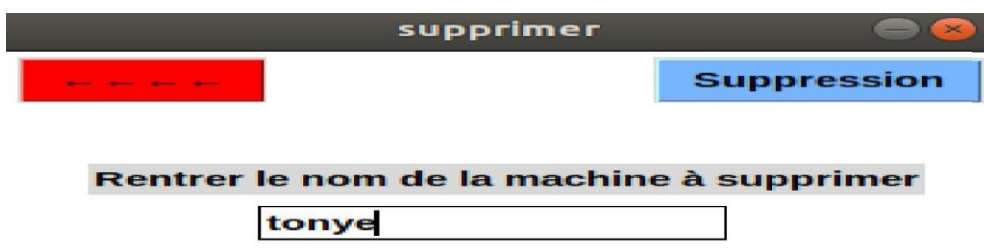
Interface de profil administrateur :



Interface Ajout d'une machine :



Interface suppression d'une machine :



Listes des machines d'un réseau



Choix de ressource de monitoring :

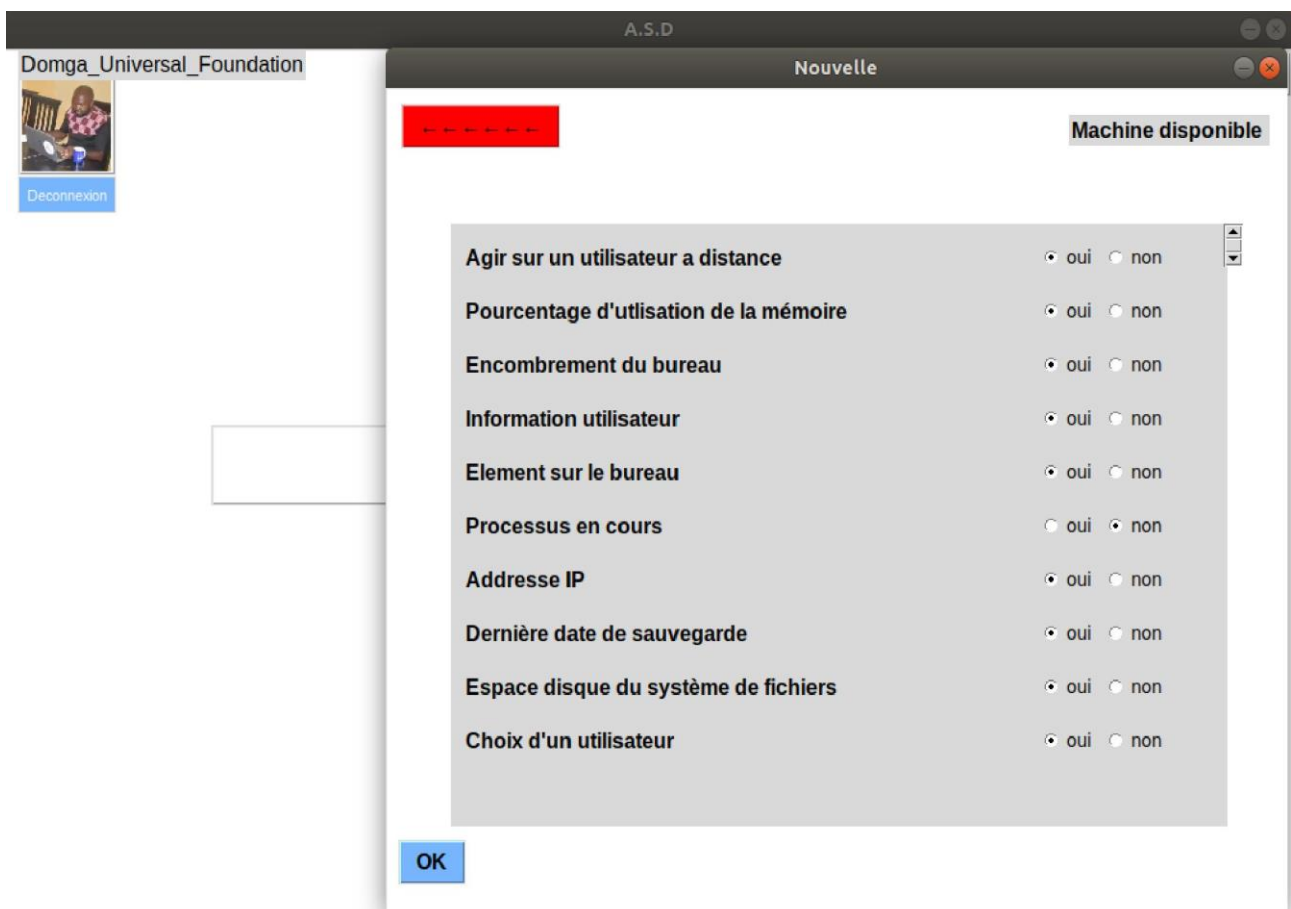


Tableau de bord de ressources supervisées :

The dashboard displays the following sections:

- Tous Connexion:**

tonye	:0	:0	Fri Jun 9 11:19 - crash (-8194+10:18)
tonye	:0	:0	Thu Jun 8 18:49 - down (16:27)
tonye	:0	:0	Thu Jun 8 18:02 - 18:14 (00:12)
tonye	:0	:0	Thu Jun 8 14:21 - down (03:17)
- Processus en cours:**

1293	whoopsie	20	0	390604	12668	11048	S	0,0	0,2	0:00.03	whoopsie
1295	root	20	0	1559140	65568	46160	S	0,0	1,1	0:00.82	dockerd
1314	kernoops	20	0	56944	420	0	S	0,0	0,0	0:00.07	kerneloops
1326	kernoops	20	0	56944	420	0	S	0,0	0,0	0:00.07	kerneloops
1341	root	20	0	218964	2700	2228	S	0,0	0,0	0:00.03	ypbind
1383	gdm	20	0	917312	50440	32872	S	0,0	0,9	0:00.15	Xwayland
1422	gdm	20	0	349276	6180	5596	S	0,0	0,1	0:00.00	at-spi-bus+
1428	gdm	20	0	49932	3692	3332	S	0,0	0,1	0:00.01	dbus-daemon
1430	gdm	20	0	220768	6824	6060	S	0,0	0,1	0:00.01	at-spi2-re+
1434	gdm	20	0	1319076	16152	12256	S	0,0	0,3	0:00.19	pulseaudio
1435	rtkit	21	1	183516	2868	2600	S	0,0	0,0	0:00.05	rtkit-daem+
1793	gdm	20	0	355816	7640	6372	S	0,0	0,1	0:00.06	ibus-daemon
1794	gdm	20	0	266008	4908	4452	S	0,0	0,1	0:00.00	xdg-permis+
1797	gdm	20	0	275192	6748	6104	S	0,0	0,1	0:00.00	ibus-dconf
1801	gdm	20	0	869988	50180	37836	S	0,0	0,9	0:00.16	ibus-x11
1805	gdm	20	0	273132	6192	5748	S	0,0	0,1	0:00.01	ibus-portal
1823	root	20	0	291292	7092	6356	S	0,0	0,1	0:00.10	boltd
1833	root	20	0	442608	66856	13016	S	0,0	1,1	0:20.01	packagekitd
1834	gdm	20	0	1020608	51632	38700	S	0,0	0,9	0:00.17	gsd-xsetti+
1841	gdm	20	0	272676	5792	5232	S	0,0	0,1	0:00.00	gsd-a11y-s+
- Charge de la memoire:**

/dev/loop7	75648	75648	0	100%	/snap/core22/750
/dev/loop14	65024	65024	0	100%	/snap/core20/1891
/dev/loop16	3456	3456	0	100%	/snap/ gnome-system-monitor/36
- Element sur le bureau:**

```

-rwxrwxr-x 1 tonye tonye 5540
-rwxr-x--x 1 tonye tonye 55408
drwxr-xr-x 3 tonye tonye 4096 déc. 4 2022 MySQL :: MySQL 8.0 F
-rw-rw-r-- 1 tonye tonye 325750 déc. 4 2022 MySQL :: MySQL 8.0
drwxrwxr-x 2 tonye tonye
drwx----- 3 tonye tonye
drwxr-xr-x 3 tonye tonye
-rwxr----- 1 tonye tonye 1673845 janv.
    
```
- Les informations identifiant:**

```

uid=1000(tonye) gid=1000(tonye) groupes=1000(tonye),4
    
```
- Contrôler (à Distance):**
 - Rédémarrer
 - Mettre en veille
 - Eteindre
 - Créer un fichier
 - Créer un dossier
 - supprimer un fichier
 - supprimer un dossier
 - renommer

Quelques cas de Test de monitoring :

The terminal shows the following commands and output:

```

Device "enp0s3" does not exist.
Monday, 12 June 2023

fichier créé
mkdir: impossible de créer le répertoire «/home/tonye/INF3216»: Le
fichier existe
Dossier créé
Fichier supprimé
Dossier supprimé
Dossier créé
renamed '/home/tonye/domga.txt' -> '/home/tonye/tonye.txt'
Renommé
fichier créé
fichier créé
renamed '/home/tonye/domga.txt' -> '/home/tonye/tonye.txt'
Renommé
    
```

The remote control interface shows the following actions:

- Rédémarrer
- Mettre en veille
- Eteindre
- domga.txt - Créer un fichier
- inf3216 - Créer un dossier
- titi.txt - supprimer un fichier
- papa - supprimer un dossier
- domga.txt - tonye.txt - renommer

Vérification des tests sur la machine distante :

The terminal shows the output of the 'ls' command:

```

tonye@tonye-HP-ProBook-6555b:~$ ls
bonjour.txt      exemples.desktop  papl          tato.txt
Bureau           Images            pipos         Téléchargements
cc.c             inf3216           projet3016   tonye.txt
cour            INF3216          Public       TPINF3015.mar
'Document 1 sans titre' Modèles          tagne        Vidéos
Documents        Musique           tail2        y
donga.txt        nyenv            tata.txt     y.pub
    
```

The remote control interface shows the following actions:

- Rédémarrer
- Mettre en veille
- Eteindre
- domga.txt - Créer un fichier
- inf3216 - Créer un dossier
- titi.txt - supprimer un fichier
- papa - supprimer un dossier
- domga.txt - tonye.txt - renommer

CONCLUSION GENERALE

Nous pouvons affirmer que ce projet a été non seulement une occasion pour nous de nous familiariser avec le milieu professionnel, mais aussi une opportunité d'acquérir des connaissances pratiques sur l'administration et la supervision de machines à distance orienté Cloud. Nous avons identifié les besoins spécifiques de ce type de système, en tenant compte des contraintes liées à l'administration à distance et à l'utilisation du Cloud. En utilisant des outils appropriés tels que les services de Cloud computing et les protocoles de gestion à distance, nous avons pu mettre en œuvre une solution qui répond aux besoins énoncés dans le cahier des charges.

Cependant, nous reconnaissons que l'administration et la supervision de machines à distances sont des domaines en constante évolution, avec de nouvelles technologies et exigences émergentes. Par conséquent, ce projet reste ouvert à toutes les améliorations futures nécessaires pour rester à jour avec les avancées technologiques et les besoins changeants.

PERSPECTIVES

En guise de perspective, ce travail reste ouvert à toutes améliorations envisageables. Comme :

- Terminer la phase de réalisation et mettre en place notre logiciel.
- Ouvrir le champ d'étude sur d'autres structures afin que cette application soit exploitable par ceux-ci et puisse répondre à d'autres besoins que nous n'avons pas spécifiés.
- Terminer la phase de correction et visualisation des résultats par les graphes.