

Université de Yaoundé 2

Institut Universitaire Catholique
Sainte Thérèse de Yaoundé

Département de Gestion des Systèmes
d'Information



The University of Yaounde 2

Catholic University Institute
Sainte Therese of Yaounde

Information Systems Management
Department

Rapport de fin de stage académique

MISE EN PLACE D'UN SYSTÈME DE STOCKAGE ET DE PARTAGE DE FICHIERS

Stage effectué du 02 Août au 10 octobre 2021 dans l'entreprise e-Robot



En vue de l'obtention du Brevet de Technicien Supérieur

Rédigé et soutenu par :

MEZUI SIANI TCHAPIE Serge-Pasker

Gestion des Systèmes d'Information

INU20SI0438

Sous l'encadrement de :

Encadreur Académique

M. ANGOUAKAYE WALWAL

Ingénieur de conception en Informatique

Encadreur professionnel

M. OSSOUBITA'A Cédric

Ingénieur Roboticien

Année Académique 2021 / 2022

DEDICACE

Je dédicace ce rapport a ma famille entière.

REMERCIEMENTS

Ce rapport de stage est le fruit de deux années de travail et de formation.

Je remercie sincèrement toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet.

Je remercie,

- Le fondateur de l'Institut Universitaire Catholique Sainte Thérèse de Yaoundé (INUCASTY) Monseigneur Jean MBARGA, Archevêque Métropolitain de Yaoundé pour m'avoir permis de passer ces années de formation dans les meilleures conditions possibles ;
- Mon encadreur académique Ingénieur ANGOUAKAYE WALWAL pour m'avoir épaulé et orienté dans l'étude de ce projet ;
- Mon encadreur professionnel et directeur de l'entreprise e-Robot, pour m'avoir accueilli dans sa structure et pour les directives et conseils qu'il m'a donné ;
- Mes camarades de classe qui m'ont soutenu et aidé dans l'élaboration et l'implémentation de ce projet.

RÉSUMÉ

Le présent rapport est issu du stage académique réalisé dans l'entreprise e-Robot. Ce stage consistait à automatiser le suivi des fichiers numérique et physique de l'entreprise. Les entreprises créent, stockent et partagent une grande quantité de document en version papier ou numérique. La manipulation de ces documents devient une tâche lourde au fil du temps et de l'accroissement des entreprises. La gestion et le suivi de ces documents est donc un réel enjeu pour elles.

Dans l'entreprise e-Robot, la gestion des documents est essentiellement manuelle. Malgré des techniques de stockage des documents mis en place par celle-ci, le partage et l'archivage des documents reste archaïque au sein de cette entreprise liée de près à l'informatique. Il a donc été question pour nous de mettre en place un système informatisé de partage, de stockage et de manipulation des fichiers de l'entreprise.

Pour atteindre cet objectif, nous avons utilisé la méthode UP (Unified Process, en anglais) pour transformer les besoins ressortis dans le cahier de charges en système logiciel. Nous avons utilisé le langage UML pour modéliser les différentes fonctionnalités de notre système. Nous avons ainsi mis en place un serveur de stockage et de partage de données sous linux et implémenté une application Web permettant d'interagir avec ce serveur dans un réseau.

Mots clé : Serveur Linux, UML, Application Web, Stockage, Messagerie Interne.

ABSTRACT

This report is the result of an academic internship carried out in the company e-Robot. This internship consisted in automating the tracking of the company's digital and physical files. Companies create, store and share a large quantity of documents in paper or digital form. The handling of these documents becomes a cumbersome task as companies grow. Managing and tracking these documents is therefore a real challenge for them.

In the e-Robot company, document management is essentially manual. Despite the document storage techniques implemented by the company, the sharing and archiving of documents remains archaic within this company closely linked to IT. We therefore decided to set up a computerized system for sharing, storing and handling the company's documents.

To achieve this objective, we used the UP (Unified Process) method to transform the requirements from the specifications into a software system. We used the UML language to model the different functionalities of our system. We set up a data storage and sharing server under Linux and implemented a web application allowing to interact with this server in a network.

Keywords: Linux server, UML, Web application, Storage, Internal messaging.

SOMMAIRE

<i>Chapitre 1 : Présentation Générale de l'entreprise</i>	3
Section I : Présentation de l'entreprise e-Robot	4
Section 2 : Accueil dans l'entreprise	6
<i>Chapitre 2 : Description du projet</i>	9
Section 1 : Contexte et Problématique	10
Section 2 : Étude de l'existant	11
Section 3 : Situation ayant conduit à ce travail	20
Section 4 : Cahier des charges	21
<i>Chapitre 3 : Méthodologie et technique</i>	28
Section 1 : Méthode de conception	29
Section 2 : Outils de conception	35
Section 3 : Conception du système	47
<i>Chapitre 4 : Résultat des tests</i>	57
Section 1 : Configuration du serveur	58
Section 2 : Programmes et Interfaces de Ralph Server	60

LISTE DES FIGURES

Figure 1 : Photo de disque dur	11
Figure 2 : Photo d'un NAS	13
Figure 3 : Architecture d'un Serveur NAS	14
Figure 4 : Le Cloud	17
Figure 5 : Organisation des tâches	24
Figure 6 : Diagramme de GANTT	24
Figure 7 : Processus Unifié	29
Figure 8 : Logo UML [4]	32
Figure 9 : Interface de Power AMC	35
Figure 10 : Interface GANTT Project	36
Figure 11 : Logo Ubuntu [5]	37
Figure 12 : Interface d'accueil d'Ubuntu Server après la connexion	38
Figure 13 : Représentation du RAID 0	40
Figure 14 : Représentation du RAID 1	41
Figure 15 : Représentation du RAID 10	41
Figure 16 : Représentation du RAID 5	42
Figure 17 : Représentation des couches du modèle LVM	43
Figure 18 : Partitionnement des disques avec LVM	44
Figure 19 : Exemple de diagramme de cas d'utilisation [6]	47
Figure 20 : Diagramme de cas d'utilisation	48
Figure 21 : Exemple de diagramme de séquence [6]	49
Figure 22 : Diagramme de séquence	50
Figure 23 : Exemple de diagramme d'état transition [6]	50
Figure 24 : Diagramme d'état transition	51
Figure 25 : Exemple de diagramme d'activité [6]	52
Figure 26 : Diagramme d'activité	53
Figure 27 : Exemple de diagramme de classe [6]	54
Figure 28 : Diagramme de classe	55
Figure 29 : Exemple de diagramme d'objet [6]	56
Figure 30 : Diagramme d'objet	56
Figure 31 : Configuration des volumes RAID	58
Figure 32 : Configuration du serveur WEB	59
Figure 33 : Page d'index	60
Figure 34 : Classe Groupe	60
Figure 35 : Interface d'accueil d'administration	62
Figure 36 : Interface d'accueil d'utilisateur standard	62
Figure 37 : Administration des groupes	63
Figure 38 : Envoi de message privé	63
Figure 39 : Logo Ralph Server	66

LISTE DES TABLEAUX

<i>Tableau 1 : Fiche Signalétique de l'entreprise e-Robot</i>	4
<i>Tableau 2 : Ressources logicielles de réseau</i>	25
<i>Tableau 3 : Ressources logicielles de développement</i>	25
<i>Tableau 4 : Ressources matérielles</i>	26
<i>Tableau 5 : Ressources humaines</i>	26
<i>Tableau 6 : Autres ressources</i>	27
<i>Tableau 7 : Récapitulatif des ressources</i>	27
<i>Tableau 8 : Outils de développement</i>	46

LISTE DES ABRÉVIATIONS

AFP: AppleShare File Protocol

CIFS: Common Internet File System

CPU : Central Processing Unit

CSS: Cascading Style Sheets

GED : Gestion Électronique de Document

HDD: Hard Disk Drive

HTML: Hypertext Markup Language

IP: Internet Protocol

LVM: Logical Volume Management

MVC: Model View Controller

NAS: Network Attached Storage

NFS: Network File System

PHP : HyperText Preprocessor

RAID: Redundant Array of Independent

RAM: Random Access Memory

SAN: Storage Area Network

SATA: Serial Advanced Technology Attachment

SCSI: Small Computer System

SQL: Structured Query Language

TCP: Transmission Control Protocol

UML: Unified Modeling Language

UP: Unified Process

INTRODUCTION GÉNÉRALE

L'entreprise e-Robot est un groupe exerçant dans le domaine de la robotique et ses sous branches, dans le développement et dans l'enseignement. Du fait de ces multiples domaines d'activité, elle enregistre chaque jour une grande quantité de documents papier et numérique. Ces documents sont gérés de façon manuelle et stocké dans un bureau dans des classeurs pour les versions physiques ou dans les ordinateurs des personnes ayant créé ces fichiers pour les versions numériques. Le partage des documents se fait essentiellement via des supports de stockage tel que des clés USB ou de bureau a bureau à l'intérieur de l'entreprise. En effet, cette manipulation des documents les expose à des risques d'altération, de perte ou de destruction et le problème de confidentialité.

L'objectif de ce projet est de mettre en place un système capable de faciliter et d'améliorer la gestion et le partage des documents dans l'entreprise. Ce système quasi autonome est capable de stocker et de partager les documents au sein de l'entreprise. Il peut fidèlement les restituer à qui de droit à tout moment. Il est ainsi capable de garantir l'intégrité et la confidentialité.

Pour atteindre ces objectifs, nous avons utilisé la méthode de Processus Unifié. Elle nous a permis après la conception du système grâce à UML, de mettre en place un serveur de stockage sous Linux permettant l'archivage des fichiers sur des disques avec redondance des données. Puis nous avons implémenté sur ce même serveur une application web qui nous permet d'accéder aux documents de façons sécurisé et de manière à contrôler les interactions Client-Serveur.

Dans la suite de notre travail, nous verrons dans le premier chapitre l'environnement dans lequel nous avons effectué notre stage académique en entreprise ainsi que le déroulement de celui-ci. Ensuite, dans le second chapitre nous verrons une description exhaustive de notre projet à travers l'étude de l'existant, la situation ayant conduit à ce travail et la réalisation du cahier de charges. Puis, dans le troisième chapitre, les méthodologies et techniques de conception qui porteront un accent sur les étapes de conception du système. Enfin le dernier chapitre portera sur les résultats des différents tests effectués sur le système finalisé.

Chapitre 1 : Présentation Générale de l'entreprise

Dans ce chapitre, nous présenterons l'entreprise et le cadre dans lequel s'est déroulé notre stage académique (Section 1). Il sera aussi question de décrire les différentes tâches qui nous ont été confiés pendant ce stage (Section 2).

Section I : Présentation de l'entreprise e-Robot

e-ROBOT ENTREPRISE est un groupe de techniciens dans différents domaines de l'informatique, exerçant principalement dans la Robotique et ses sous-branches : domotique, intelligence artificielle, robotique industrielle, mécatronique ; dans le développement web et mobile, dans les installations réseaux, dans la conception, la mise en place et la maintenance des systèmes d'informations et dans l'enseignement.

I.1. Fiche Signalétique

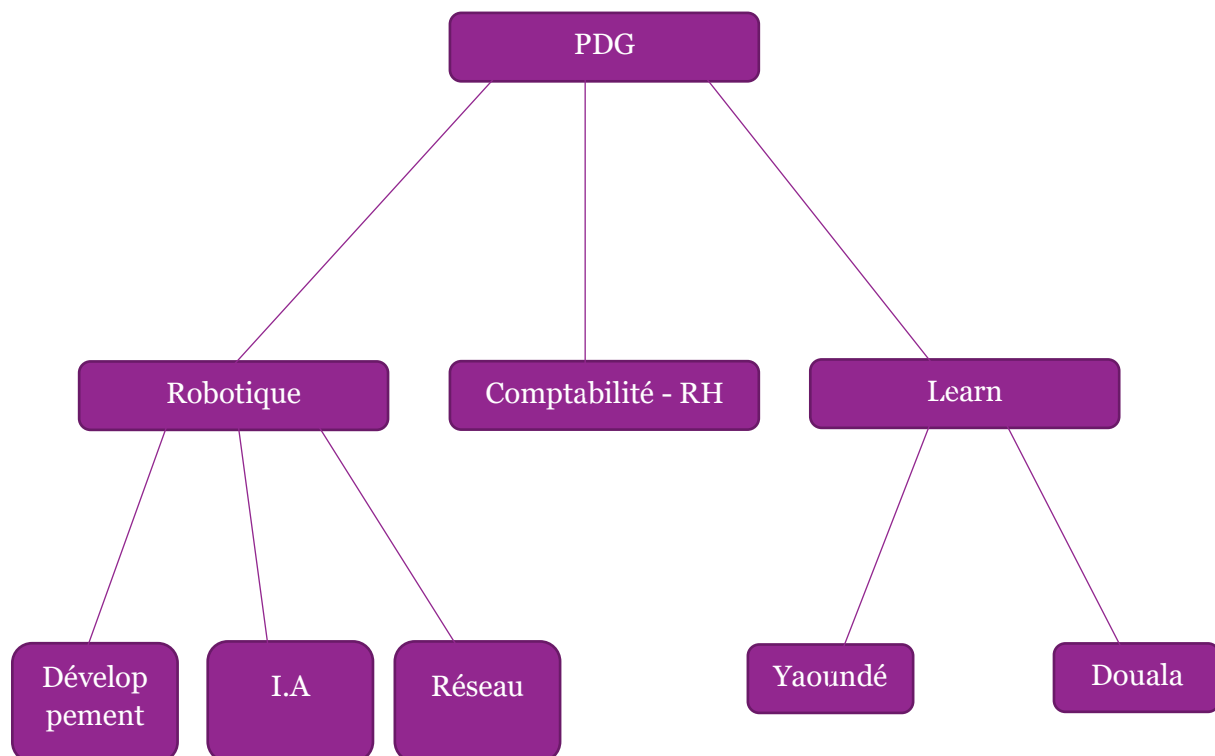
<i>Descriptif</i>	<i>Description</i>
<i>Dénomination</i>	Établissement e-Robot
<i>Statut</i>	Établissement
<i>Nombre du personnel</i>	17 hommes, 3 femmes
<i>Contact public</i>	+237 656504909
<i>Zone d'intervention</i>	Cameroun
<i>Activité principale</i>	Informatique
<i>Directeur</i>	M. Cédric OSSOUBITA'A
<i>Activités secondaire</i>	Formations professionnelles
<i>Défis majeur</i>	Relever l'image du Cameroun sur le plan des Technologies de l'information et de la communication

Tableau 1 : Fiche Signalétique de l'entreprise e-Robot

I.2. Partenaires

- MINJEC
- CMPJ MFOUNDI
- CMPJ WOURI
- CMPJ Littoral
- CMPJ Mvilla
- AchD Entreprise,
- ETS Émergence
- ETS La diversité
- AUTO-ECOLE Afrique
- AUTO-ECOLE Panafricaine
- AUTO-ECOLE Mexicaine
- RJC
- C.O. Tech Food
- AUTO6ECOLE La Prudence

I.3. Organigramme



Section 2 : Accueil dans l'entreprise

II.1. Présentation et Déroulement du stage

Le stage dans l'entreprise e-Robot commence le 2 Aout 2021. Nous avons été présenté au personnel et aux différents chefs des départements. Une réunion a été faite avec les chefs de départements et les encadreurs professionnels dans le but d'élaborer le plan de travail dans l'entreprise en fonction de nos différentes spécificités.

Ce plan résumait les départements dans lesquels nous devrions travailler et la période à faire dans ces différents départements.

II.2. Plan d'affectation

Le stage dans l'entreprise e-Robot a duré exactement neuf (9) semaines.

Les deux (2) premières semaines se sont déroulées dans le département de Design. Les cinq (5) semaines suivantes dans le département de développement et les deux (2) dernières semaines à l'assistante, chargée des Ressources Humaines.

II.3. Chronologie

II.3.1. Première affectation : Département de Design

Le travail qui nous avait été assigné consistait en la réalisation de maquettes et prototypes de sites web. On avait donc la tache de réaliser la maquette du site officiel, public de l'entreprise, e-Robot.

Le maquettage se faisait d'un premier temps à l'aide d'un tableau et un feutre et après validation, il s'agissait de le matérialiser dans un ordinateur à l'aide du logiciel XD de la suite Adobe.

Cet outil nous permet non seulement de dessiner le design du futur site mais aussi de visualiser les interactions avec l'utilisateur : animation des boutons, clic, survol, scroll, rotation d'image...

II.3.2. Seconde affectation : Département de Développement

Dans ce département sont traités les projets de l'entreprise qui sont en phase d'implémentation. Il est constitué essentiellement de développeurs et d'analystes. Logiquement il nous avait été confié d'implémenter la maquette du site de l'entreprise.

Le site a été développé dans les langages HTML, CSS, JavaScript avec l'éditeur de code Visual Studio Code de Microsoft. Nous avons réalisé les interfaces, les responsives et les intégrations web.

Compte tenu du temps qui nous avait été alloué dans ce département, nous n'avons pas pu réaliser entièrement que le frontend du site. Pour la partie serveur, nous avons déployé les bases de données et quelques migrations.

II.3.3. Troisième affectation : Département des ressources Humaines

Dans ce département, il s'agissait d'assister le Directeur des Ressources Humaines dans ses tâches quotidiennes. Celle de l'archivage et le stockage des documents.

Les documents administratifs et les documents fonctionnels de l'entreprises étaient stockés dans un local accessible par tout le monde et les documents numériques étaient quant à eux stockés dans le disque dur du Directeur des Ressources Humaines. Pour partager ces dossiers avec le personnel il fallait les mettre dans une clés USB dont on ne connaissait pas la provenance. La gestion des documents se résumait à cela.

II.2. Taches effectuées

Dans cette entreprise nous avons réalisé les tâches de maquettage et de prototypage des designs de site web, implémentation de site web ainsi que la gestion des ressources de l'entreprise.

II.4. Difficultés rencontrées

Pendant la durée de notre stage, nous avons rencontré les difficultés suivantes :

- Les nouveaux outils de programmation ;
- Le design ;
- Gestion de la paperasse.

II.5. Bénéfices

Étant donné la difficulté et la diversité des tâches effectuées dans cette entreprise, ce stage a eu des apports en connaissances sur nous :

- Nous avons appris le travail en entreprise ;
- Nous avons appris les canevas d'implémentation d'un projet ;
- Nous avons appris à mieux gérer le temps et les coûts de production.

Chapitre 2 : Description du projet

Dans ce chapitre, nous ressortirons tout d'abord le contexte et la problématique qui a abouti à ce projet (Section 1), puis nous étudierons l'état de l'art et de l'existant (Section 2), et nous finirons par l'élaboration du cahier de charges de notre projet (Section 3).

Section 1 : Contexte et Problématique

I.1. Contexte

Dans les entreprises, la gestion des documents et du stockage en général n'est pas toujours évidente. Les petites et moyennes entreprises sont confrontées à la mauvaise gestion de leurs documents administratifs et surtout les documents privés. En plus de cela le partage de ces documents en format physique ou en format numérique se faisant dans des conditions insécurisées expose ces documents à un risque de perte ou de vol. La vulnérabilité dans la gestion de ces documents expose le système d'information de l'entreprise.

I.2. Problématique

L'entreprise stocke une grande quantité de documents en version physique dans leurs locaux et en version numérique dans les disques dur de leurs ordinateurs, ce qui peut causer les problèmes de :

- Perte des données sensible ;
- Mauvais suivi des documents ;
- Viol de confidentialité ;
- Accès des données.

Section 2 : Étude de l'existant

II.1. État de l'art

Depuis le début de l'humanité, l'être humain a toujours cherché un moyen de transmettre des connaissances à travers les époques. Des peintures rupestres aux livres, en passant par les gravures sur la roche ou des rouleaux de papyrus, à l'ère du numérique, les supports de stockage de données ont évolué.

Les entreprises constituant un système doivent s'armer d'outils informatiques pour stocker, partager et sécuriser efficacement leurs données numériques.

Pour ce faire, les entreprises ont les choix entre plusieurs moyens de stockage de données qui existent.

II.1.1. Le stockage de masse ou stockage physique

Au fil des générations les supports de stockage de masse ont évolué, faisant place à des versions plus performantes, plus petites et plus rapides.



Figure 1 : Photo de disque dur

Les supports de stockage les plus courants de nos jours appartiennent à la troisième, quatrième, et cinquième génération (disque compact, DVD, Blu-ray, clé USB, carte SD, carte micro SD, support SSD).

Les disques durs, les clés USB et les disques optiques sont déjà très sollicités en entreprise.

➤ Avantages :

- ✓ Faible coût d'acquisition ;
- ✓ Simple à utiliser ;
- ✓ Aucune connexion internet n'est nécessaire ;
- ✓ Capacité de stockage élevé.

➤ Inconvénients :

- ⊗ Pannes matérielles entraînant la perte intégrale des données ;
- ⊗ Piratage physique (vol, destruction, ...).

II.1.2 Le stockage Réseau

Le stockage réseau permet de rendre les données stockées accessibles par tous les éléments du réseau. Les éléments du réseau étant des périphériques, les données sont stockées à un ou plusieurs endroits du réseau.

Comme technologie de stockage en réseau nous avons le NAS qui repose sur une architecture de stockage en mode fichier et comme réseau de stockage nous avons le SAN qui se repose sur un réseau de stockage et les échanges de données s'effectue en mode bloc.

II.1.2.1 Le NAS

NAS (ou Network Attached Storage) est l'architecture de stockage qui rend les données stockées plus accessibles aux périphériques réseau. Il offre aux réseaux un point d'accès unique pour le stockage qui intègre des fonctionnalités de sécurité, de gestion et de tolérance aux pannes.



Figure 2 : Photo d'un NAS

a) Matériels et logiciels

Un logiciel de stockage préconfiguré est installé sur du matériel spécifique. Le matériel à la base du stockage en réseau est généralement appelé boîtier, unité ou serveur de stockage en réseau. Il s'agit d'un serveur contenant des disques de stockage, des processeurs et de la mémoire RAM et son propre système d'exploitation.

La différence principale entre un NAS et un serveur de stockage généraliste se situe au niveau du logiciel. Le logiciel du NAS est déployé sur un système d'exploitation léger généralement intégré au matériel. Les serveurs de stockage généralistes disposent d'un système d'exploitation complet qui envoie et reçoit des milliers de requêtes chaque seconde, dont une fraction seulement concerne le stockage. À l'opposé, le boîtier NAS n'envoie et ne reçoit que deux types de requêtes : stockage des données et partage de fichiers.

b) Protocol

Un boîtier de stockage en réseau est formaté avec des protocoles de transfert de données qui régissent l'envoi de données entre plusieurs périphériques. Les clients peuvent accéder à ces protocoles via un commutateur, c'est-à-dire un serveur central qui connecte tous les éléments du réseau et achemine les requêtes.

Sur un réseau, plusieurs protocoles peuvent être utilisés, mais les deux plus importants sont les suivants : le protocole IP (Internet Protocol) et le protocole TCP (Transmission Control Protocol). Le protocole TCP permet de regrouper les données en paquets avant de les envoyer via le protocole IP. Pour simplifier les choses, les paquets TCP sont comparables à des fichiers ZIP et le protocole IP à des adresses e-mail.

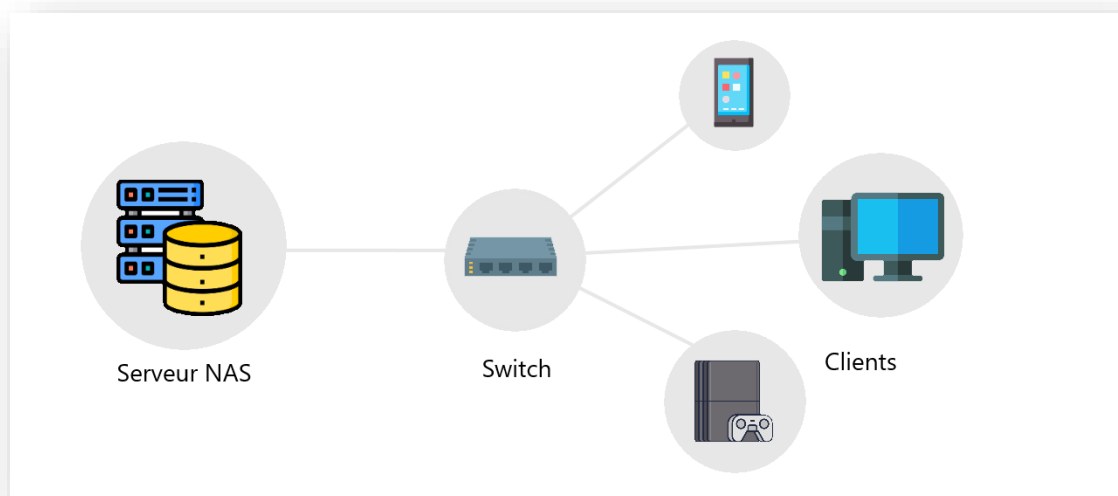


Figure 3 : Architecture d'un Serveur NAS

Le serveur NAS intègre le support de multiples systèmes de fichiers réseau, tels que Common Internet File System (CIFS) protocole de partage de Microsoft et de Samba, Network File System (NFS) qui est un protocole de partage de fichiers Unix, ou encore AFP (AppleShare File Protocol) qui est l'équivalent pour la technologie Apple. Une fois connecté au réseau, il peut jouer le rôle de plusieurs serveurs de fichiers partagés.

c) Avantages

- **Évolutivité** : pour augmenter la capacité de stockage d'un NAS, il suffit d'y ajouter des disques durs. Pas besoin de mettre à niveau ni de remplacer des serveurs, et encore moins de désactiver le réseau.
- **Performances** : comme le NAS est un système de fichiers, les autres périphériques du réseau n'ont pas besoin de se charger du partage des fichiers. De plus, le NAS est souvent configuré pour une utilisation bien précise (stockage de Big Data ou de contenus multimédias, par exemple), ce qui permet d'obtenir de meilleures performances.
- **Configuration simple** : les architectures NAS sont souvent accompagnées de scripts simples ou encore fournies sous la forme d'applications préinstallées avec un système d'exploitation rationalisé, ce qui réduit considérablement le temps de configuration et de gestion du système.
- **Accessibilité** : tous les périphériques en réseau ont accès au NAS.
- **Tolérance aux pannes** : il est possible de formater le NAS de sorte qu'il prenne en charge des disques répliqués, un RAID (Redundant Array of Independent Disks) ou le codage à effacement afin d'assurer l'intégrité des données.

II.1.2.2 Le SAN

a) Matériels et logiciels

Les SAN sont construits dans le but de fournir de l'espace disque rapide et fiable. La technologie la plus répandue pour y parvenir est l'utilisation du protocole Fibre Channel, qui permet d'atteindre des débits élevés. Cette technologie induit la mise en œuvre de switch ou de directors (gros switches de plus de 64 ports, garantissant une disponibilité maximale et une évolutivité par ajout de cartes contenant des ports de connexion). Le réseau de stockage ainsi constitué est appelé un fabric.

b) Avantages

- **Évolutivité** : L'espace disque n'est plus limité par les caractéristiques des serveurs, et est évolutif à volonté par l'ajout de disques ou de baies de stockage sur le SAN. L'espace de stockage physique mutualisé pour les serveurs permet d'optimiser la gestion des disques, et de rendre plus aisées les sauvegardes de données.
- **Sauvegarde** : Les ressources de stockage ainsi mutualisées (SAN ou NAS) donnent la possibilité de mettre en œuvre des fonctions de réplication (copie de données synchrone ou asynchrone entre deux baies) et de snapshot.
- **Accessibilité** : Le SAN peut fonctionner dans un environnement complètement hétérogène : les serveurs Unix, Windows, Netware... peuvent tous rejoindre le SAN.
- **Tolérance des pannes** : Le SAN peut assurer la redondance du stockage, c'est-à-dire l'accessibilité au système de stockage en cas de panne de l'un de ses éléments, en doublant au minimum chacun des éléments du système (haute disponibilité).
- **Vitesse** : Le SAN utilise son propre réseau dédié au stockage.

II.1.3 Le stockage Cloud

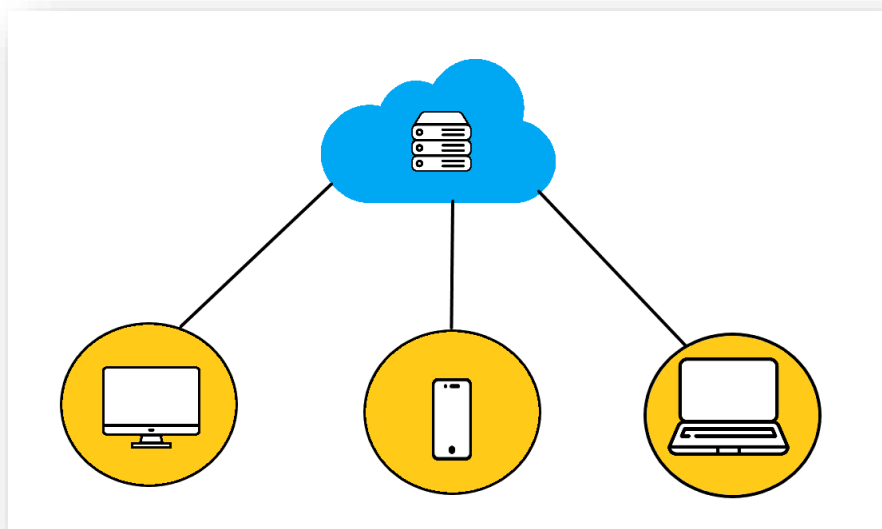


Figure 4 : Le Cloud

Le cloud (nuage, en anglais) est un moyen de sauvegarde des données numériques permettant d'enregistrer et conserver les données sur des serveurs mutualisés ou privés via internet.

Les documents stockés sur un serveur cloud sont accessibles partout et en permanence, depuis n'importe quel appareil pouvant accéder à internet. C'est donc une solution qui répond aux besoins de mobilité de plus en plus élevé des professionnels et des équipes de travail. C'est la raison pour laquelle le cloud est souvent au service du télétravail et du travail collaboratif en entreprise

➤ Avantages :

- ✓ Les données sont accessibles partout ;
- ✓ Prix abordables ;
- ✓ Gestion cohérente des données ;
- ✓ Une reprise après sinistre fiable.

➤ Inconvénients :

- ⊗ Assez couteux ;
- ⊗ Système de stockage ne vous appartient pas ;
- ⊗ Vulnérabilité des Data Center.

II.2 Étude de l'existant

II.2.1 Description du système de gestion des documents

a) Gestion des documents physiques

Les documents sont stockés dans un bureau en version physique, rangés dans différents classeurs en fonction de leur genre.

Tous les employés ont accès à ce bureau. Mais les documents sensibles sont stockés dans le bureau du PDG ou de celui du Directeur des Ressources Humaines.

b) Gestion des documents numériques

Les documents en version numérique quant à eux sont stocké sur le poste de travail du propriétaire ou de l'auteur.

Plusieurs versions de ces documents existent sur plusieurs machines et surtout sur les supports de stockages.

c) Partage des fichiers dans l'entreprise

Pour partager des documents en version physique (pour une signature ou une validation), ils sont transportés vers le destinataire. Pour les documents en version numérique, ils sont soit stocké dans des support tel que des clés USB ou des disques durs, soit envoyé par mail. Ces clés sont transportées à travers tout le personnel et en cas de perte, c'est l'intégralité des données qui sont perdu.

II.3 Critique de l'existant

II.3.1 Avantages

- ✓ Fichiers rangés dans des dossiers ;
- ✓ Fichiers accessibles par tous ;
- ✓ Documents facilement transportables.

II.3.2 Inconvénients

- ⊗ Documents accessibles à n'importe qui ;
- ⊗ Mauvais suivi des documents ;
- ⊗ Risque de perte de données trop élevé ;
- ⊗ Perte intégrale des données en cas de sinistre.

Section 3 : Situation ayant conduit à ce travail

III.1 Description

La gestion des documents de l'entreprise e-Robot est essentiellement faite de façon manuel, avec l'utilisation de techniques de Gestion Électronique des Documents (GED) presque archaïque.

Les documents sont stockés et archivés dans un local accessible par tous et en une seule copie. Ce qui laisse une grande vulnérabilité et augmente le risque de perte de données par des litiges, des vols ou même de piratage. Face à ce tas de problèmes qu'engendre ce système, nous avons mis sur place une solution.

III.2 Proposition de solution

La solution proposée pour ce problème de gestion de documents est l'application Ralph Server, un serveur de fichier permettant de stocker et partager les documents de l'entreprise lié à un logiciel permettant de manipuler ces documents.

Il s'agira premièrement de mettre en place un serveur de fichiers sous linux avec un système de RAID et de sauvegarde des données sur un serveur distant.

Et deuxièmement d'implémenter un logiciel de gestion de données avec une interface homme-machine.

Section 4 : Cahier des charges

IV.1 Contexte et justifications

IV.1.1 Contexte

Dans les entreprises, la gestion des documents et de l'archivage en général n'est pas toujours évidente. Les petites et moyennes entreprises sont confrontées à la mauvaise gestion de leurs documents administratifs et surtout les documents privés. En plus de cela le partage de ces documents en format physique ou en format numérique se faisant dans des conditions insécurisées expose ces documents à un risque de perte ou de vol. La vulnérabilité liée à l'exploitation de ces documents rend aussi vulnérable de système d'information de l'entreprise.

IV.1.2 Problème

L'entreprise stocke une grande quantité de documents en version physique dans leurs locaux et en version numérique dans les disques dur de leurs ordinateurs, ce qui peut causer les problèmes de :

- Perte des données sensible ;
- Mauvais suivi des documents ;
- Vulnérabilité des données ;
- Failles de sécurité des données.

IV.2 Objectifs

IV.2.1 Objectif globale

L'objectif global de notre projet est d'implémenter un outil permettant d'assurer la sécurité, l'intégrité, le stockage numérique des documents de l'entreprise dans le but de manager le partage de ces données dans un réseau.

IV.2.2 Objectif spécifique

Les objectifs spécifiques de notre projet sont :

- Permettre l'accès aux données stockés sur le serveur ;
- Rendre les données disponibles à qui de droit à tout moment ;
- Faciliter le partage des documents au sein de l'entreprise ;
- Faciliter la communication dans l'entreprise.

IV.3 Besoins

IV.3.1 Besoins fonctionnels

Les besoins fonctionnels attendus par le système sont :

- Permettre à l'administrateur d'ajouter, de supprimer ou de modifier des documents ;
- Permettre à l'administrateur de savoir qui a ajouté un document et quand ;
- Permettre aux utilisateurs d'ajouter des documents selon les droits définis ;
- Archiver les documents ajoutés ;
- Sauvegarder les documents dans un disque de sauvegarde ;
- Permettre aux utilisateurs lambda de consulter les documents s'ils en ont les droits.

IV.3.2 Besoins non fonctionnels

Les besoins non fonctionnels sont des besoins de sécurité qui permettent de garantir l'intégrité et la non-répudiation des données.

- Définir des droits d'accès aux fichiers ;
- Définir un suivi des consultations des fichiers.

IV.4 Livrables

On appelle livrable le résultat tangible d'une production réelle, mesurable attendue par le client final. Au terme de la période destiné au développement de Ralph Server, les livrables seront les suivants :

- Une Application ;
- Un Serveur ;
- Un manuel d'utilisation.

IV.5 Planification et estimation des coups

IV.5.1 Planification du projet

En gestion de projet informatique la planification peut être définie comme étant le processus allant du découpage du projet en un ensemble de tâches à réaliser, de l'estimation de la durée jusqu'à l'ordonnancement des différentes tâches à effectuer afin de terminer le projet dans les délais.

Le projet Ralph Server se découpe en deux parties, une partie Réseau, qui consiste à la conception d'un serveur de stockage sous linux et une partie Développement, qui consiste à l'implémentation d'une application de gestion des fichiers sur ce serveur.

La conception de ce projet a donc une durée de trois (3) mois dont un pour la conception du réseau et deux pour la mise en place du système de gestion. Le projet se découpe donc sur les différentes phases :

- Analyse et critique de l'existant ;
- Conception du serveur ;
- Implémentation de l'application ;
- Connexion Client-Serveur et Test ;
- Déploiement.

GANTT project		
Nom	Date de début	Date de fin
• Analyse et critique de l'existant	04/10/2021	26/10/2021
• Conception du serveur	27/10/2021	15/11/2021
• Implémentation de l'application	16/11/2021	17/12/2021
• Connexion Client-Serveur et Test	20/12/2021	06/01/2022
• Déploiement	07/01/2022	13/01/2022

Figure 5 : Organisation des tâches

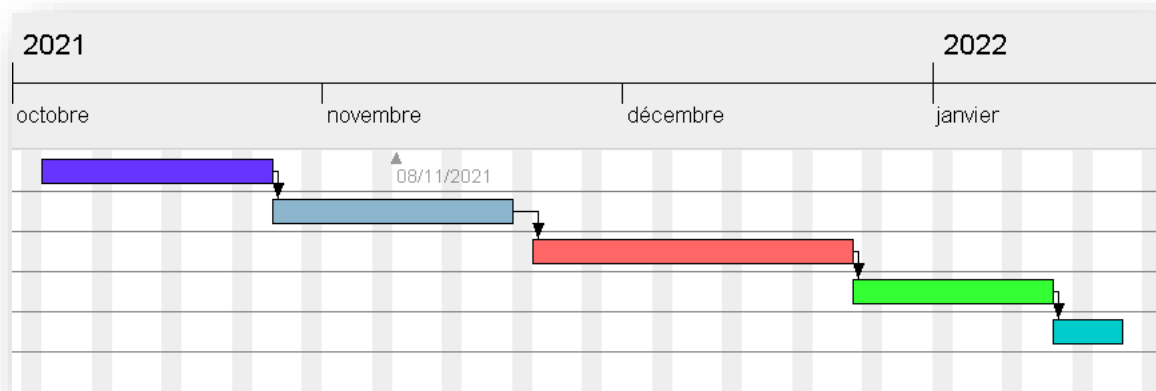


Figure 6 : Diagramme de GANTT

IV.5.2. Estimation des coûts

En estimant les coûts, on veut savoir si le projet sera rentable et s'il est vraiment nécessaire de réaliser le projet. Ainsi pour le réaliser on aura besoin de ressources logiciels, matériels et humaines.

a) Ressources logicielles :

Réseau		
Logiciels	Rôle	Prix (FCFA)
Ubuntu Server	Système d'exploitation Linux	Gratuit
Apache Server	Serveur HTTP	Gratuit
Maria DB	Système de Gestion de Base de Donnée	Gratuit
PHP Server	Serveur de traitement de données	Gratuit
Total		0 FCFA

Tableau 2 : Ressources logicielles de réseau

Développement		
Logiciels	Rôle	Prix (FCFA)
Microsoft Visual Studio Code	Éditeur de code	Gratuit
Navigateur Web	Interaction avec le serveur	Gratuit
Total		0 FCFA

Tableau 3 : Ressources logicielles de développement

b) Ressources matérielles :

Réseau & Développement			
Outils	Prix Unitaire (FCFA)	Quantité	Prix Total (FCFA)
Disque dur HDD 2To	30 000	2	60 000
Barette RAM 2Go	15 000	2	30 000
Bloc Serveur	120 000	1	120 000
Moniteur	15 000	1	15 000
Clavier	5 000	1	5 000
Scanner Professionnel	100 000	1	100 000
Routeur	100 000	1	100 000
Total			430 000

Tableau 4 : Ressources matérielles

c) Ressources humaines

Main d'œuvre	Quantité	Prix	Durée (jour)	Total
Concepteur	1	-	-	-
Développeur	1	100 000	30	3 000 000
Administrateur Système	1	100 000	30	3 000 000
Testeur	1	1 500 000	10	150 000
Total				6 150 000

Tableau 5 : Ressources humaines

d) Autres ressources

Ressources	Prix (FCFA)	Durée (jour)	Total
Connexion internet	10 000	72	30 000
Transport	500	72	36 000
Total			66 000

Tableau 6 : Autres ressources

e) Récapitulatif

Ressources	Montant total (FCFA)
Ressources Logiciels	0
Ressources Matériels	430 000
Ressources Humaines	6 150 000
Autres Ressources	66 000
Total	6 646 000

Tableau 7 : Récapitulatif des ressources

Chapitre 3 : Méthodologie et technique

Dans le but de mieux comprendre comment notre système fonctionne et comment nous l'avons conçu, ce chapitre traitera des différentes méthodes et techniques que nous avons utilisé pour le concevoir. Ainsi nous verrons d'abord les méthodes de conception (Section 1). Ensuite les outils et techniques de conception (Section 2). Enfin les différents diagrammes permettant de mieux représenter le système (Section 3).

Section 1 : Méthode de conception

I.1. Processus Unifié ou UP

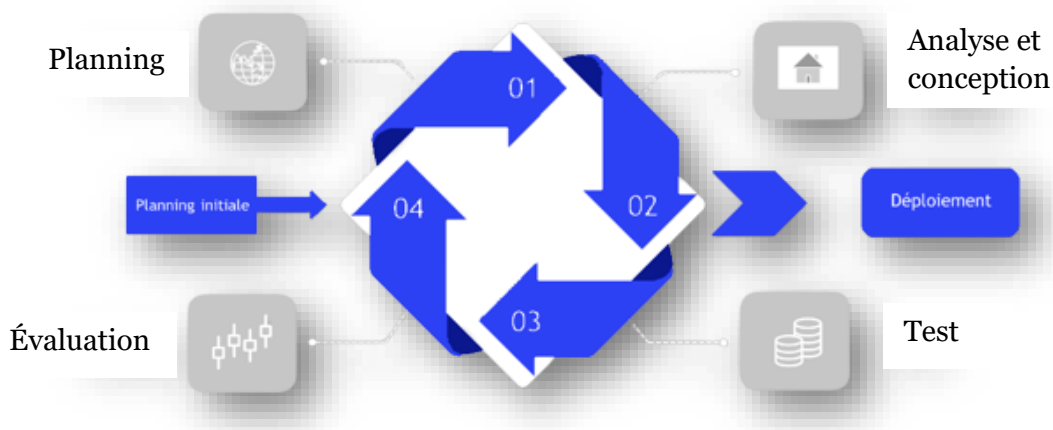


Figure 7 : Processus Unifié

Le processus unifié est un processus de développement logiciel : il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel. L'objectif principal d'un système logiciel est de rendre service à ses utilisateurs. Il faut par conséquent bien comprendre les désirs et les besoins des futurs utilisateurs. Le processus de développement sera donc centré sur l'utilisateur. Le terme utilisateur ne désigne pas seulement les utilisateurs humains mais également les autres systèmes. C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise. L'utilisateur représente donc une personne ou une chose dialoguant avec le système en cours de développement.

a) Méthodologie

La méthodologie de UP peut être représentée par ses caractéristiques, et elles sont diverses. En effet, UP est tout d'abord une méthode de développement logicielle construite sur UML. Il est également itératif et incrémental, centré sur

l'architecture (fonctionnelle, logique, matérielle), guidé par le cas d'utilisation UML, axé sur l'utilisateur et piloté par les risques.

- UP est itératif et incrémental : le projet est découpé en itérations ou étapes de courte durée qui permettent de mieux suivre l'avancement globale. À la fin de chaque itération une partie exécutable du système finale est produite, de façon incrémentale (par ajout).
- UP est centré sur l'architecture : tout système complexe doit être décomposé en partie modulaire afin d'en faciliter la maintenance et l'évolution. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML, et pas seulement documentée en texte.
- UP est guidé par les cas d'utilisation d'UML : le but principal d'un système d'informatique est de satisfaire les besoins de client. Le processus de développement sera donc axé sur l'utilisateur. Le cas d'utilisation permet d'illustrer ces besoins. Ils détectent puis décrivent les besoins fonctionnels et leur ensemble constitue le modèle de cas d'utilisation qui dicte les fonctionnalités complètes du système.
- UP est piloté par les risques : les risques majeurs du projet doivent être identifiés au plus tôt mais surtout levés le plus rapidement. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.

b) Objectifs

L'objectif d'un processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques. UP est un ensemble de principes génériques adaptés en fonction des spécificités des projets.

L'architecture bidirectionnelle : UP gère le processus de développement par deux axes :

L'axe vertical : Il représente les principaux enchaînements d'activités, qui regroupent les activités selon leur nature. Cette dimension rend compte l'aspect statique du processus qui s'exprime en termes de composants, de processus, d'activités, d'enchaînements, d'artefacts et de travailleurs.

L'axe horizontal : Il représente le temps et montre le déroulement du cycle de vie du processus ; cette dimension rend compte de l'aspect dynamique du processus qui s'exprime en termes de cycles, de phases, d'itérations et de jalons.

c) Rôle

Les rôles de UP peuvent être assimilés aux activités que l'on peut mener avec l'aide de cette méthode de développement. Nous pouvons ainsi citer :

i) Expression des besoins :

L'expression des besoins comme son nom l'indique, permet de définir les différents besoins existants :

- Inventorier les besoins principaux et fournir une liste de leurs fonctions ;
- Recenser les besoins fonctionnels (du point de vue de l'utilisateur) qui conduisent à l'élaboration des modèles de cas d'utilisation ;
- Appréhender les besoins non fonctionnels (techniques) et livrer une liste des exigences.

Le modèle de cas d'utilisation présente le système du point de vue de l'utilisateur et représente sous forme de cas d'utilisation et d'acteur, les besoins du client

ii) L'analyse

L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences du client. Il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution.

Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structures sous une forme qui facilite la compréhension (scénarios), la préparation (définition de l'architecture), la modification et la maintenance du futur système. Il s'écrit dans le langage des développeurs et peut être considéré comme une première ébauche du modèle de conception.

iii) Conception

La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et au système d'exploitation. Elle détermine les principales interfaces et les transcrit à l'aide d'une notation commune. Elle constitue un point de départ à l'implémentation : Le travail d'implémentation en sous-système.

I.2. Le Langage UML



Figure 8 : Logo UML [4]

Le langage UML ou Unified Modeling Language en anglais, est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il permet de modéliser en conception orientée objet car elle implémente bien tous ces concepts.

Depuis sa version 2.3, UML compte quatorze (14) diagrammes, et une subdivision en deux (02) grandes catégories :

- **Diagrammes de structure ou diagrammes statiques**

Les diagrammes de structure ou diagrammes statiques rassemblent :

- **Diagramme de classes** : représentation des classes intervenant dans le système.
- **Diagramme d'objets** : représentation des instances de classes (objets) utilisées dans le système.
- **Diagramme de composants** : représentation des composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données...)
- **Diagramme de déploiement** : représentation des éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage...) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent entre eux.
- **Diagramme des paquets** : représentation des dépendances entre les paquets (un paquet étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML), c'est-à-dire entre les ensembles de définitions.
- **Diagramme de structure composite** : représentation sous forme de boîte blanche des relations entre composants d'une classe.
- **Diagramme de profils** : spécialisation et personnalisation pour un domaine particulier d'un méta-modèle de référence d'UML (depuis UML 2.2).

- **Diagrammes de comportement ou diagrammes dynamiques**

Les diagrammes de comportement rassemblent :

- **Diagramme des cas d'utilisation** : représentation des possibilités d'interaction entre le système et les acteurs (intervenants extérieurs)

au système), c'est-à-dire de toutes les fonctionnalités que doit fournir le système ;

- **Diagramme états-transitions** : représentation sous forme de machine à états finis du comportement du système ou de ses composants ;
- **Diagramme d'activité** : représentation sous forme de flux ou d'enchaînement d'activités du comportement du système ou de ses composants.
- **Diagramme de séquence** : représentation de façon séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.
- **Diagramme de communication** : représentation de façon simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.
- **Diagramme global d'interaction** : représentation des enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences.
- **Diagramme de temps** : représentation des variations d'une donnée au cours du temps.

UML étant un langage formel et normalisé, il est un bon support de communication dans l'élaboration d'une solution informatique, sa polyvalence et sa souplesse font de lui un langage universel.

Section 2 : Outils de conception

I.1. Outils

II.1.1. Power AMC

Power AMC est un logiciel de conception créé par la société SAP qui permet de modéliser les traitements informatiques et leurs bases de données associées. Il permet de modéliser avec la méthode de modélisation MERISE et le langage UML en utilisant des objets dont l'interaction produit certaines opérations et qui constituent dans l'ensemble un système d'information.

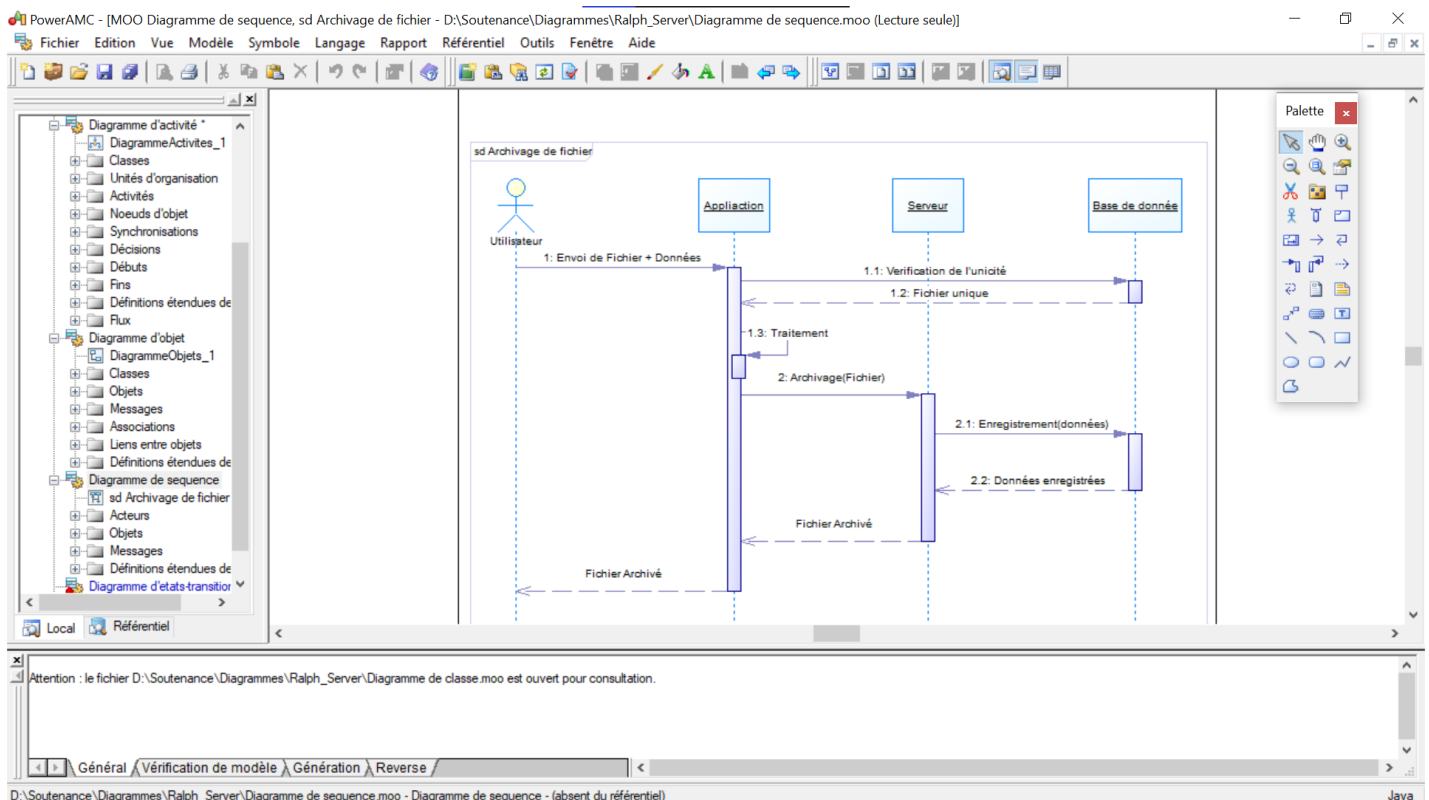


Figure 9 : Interface de Power AMC

Il permet de répliquer des objets depuis une base de données source vers un ou plusieurs moteurs de replication ou bases de données. Il permet aussi de créer une structure de programme orienté objet en Java dérivé de sa modélisation en UML

ou de générer une base de données et ses tables associées depuis sa modélisation à partir de MERISE.

II.1.2. GANTT Project

GANTT Project est un logiciel libre qui permet de planifier un projet à travers la réalisation de diagrammes de GANTT ainsi que des diagrammes de ressources et des réseaux PERT.

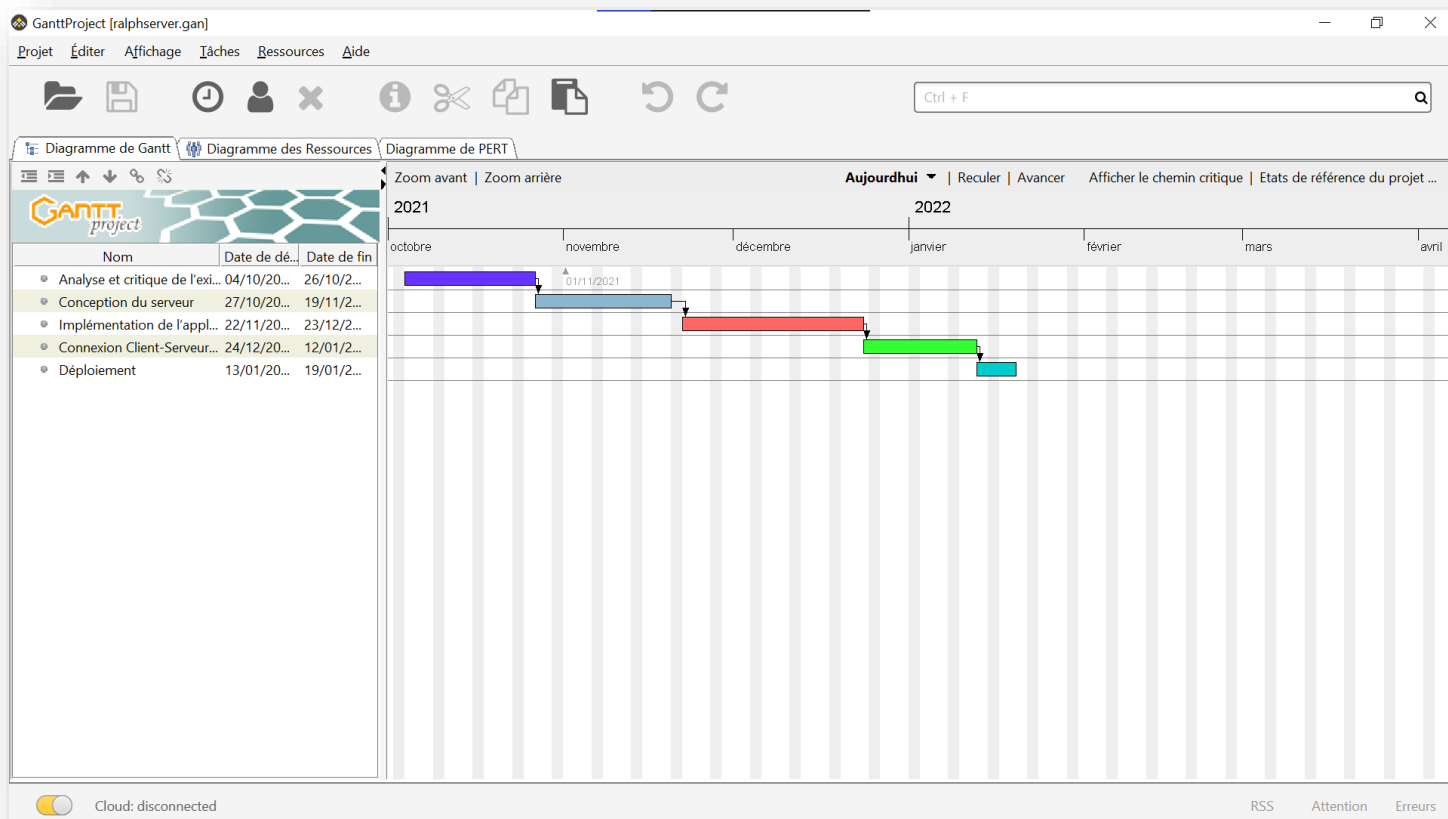


Figure 10 : Interface GANTT Project

Il permet de décrire la structure de découpage du projet (en anglais « Work Breakdown Structure ») axée sur les tâches, du travail que l'équipe de projet doit exécuter pour atteindre les objectifs du projet et produire les livrables voulus. Il permet aussi de lier de multiples précédences entre les tâches, de visualiser le

chemin critique, et ainsi l'exportation en plusieurs formats (HTML, PDF, CSV, PNG ou JPEG).

II.2. Réseau et Administration système

Les outils coté serveur sont toutes les méthodes et techniques qui vont nous permettre d'administrer le système, les disques de stockage ainsi que les différents accès réseau.

II.2.1. Ubuntu Server



Figure 11 : Logo Ubuntu [5]

Ubuntu Server fait partie de l'ensemble plus large de produits et de systèmes d'exploitation Ubuntu développés par Canonical Ltd. Ubuntu Server est un ajout spécifique qui diffère un peu de la version bureau d'Ubuntu, afin de faciliter l'installation sur les serveurs.

Les outils Ubuntu, qui sont des logiciels open source, sont des alternatives à divers types de produits sous licence. Le mot « Ubuntu » vient d'un terme sud-africain désignant « l'unité et le partage ».

La version serveur d'Ubuntu est une version qui embarque uniquement les fonctionnalités principales dont ont besoins les serveurs, il est donc dépourvu d'interface graphique, une bonne connaissance du terminal et des commandes Shell est requise pour l'utiliser.

```
Ubuntu 20.04.3 LTS vm-server tty1
vm-server login: tom
Password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-91-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of mar. 15 mars 2022 09:41:00 UTC

System load:  0.7               Processes:            116
Usage of /:   46.3% of 9.78GB   Users logged in:      0
Memory usage: 20%              IPv4 address for enp0s3: 10.0.2.15
Swap usage:   0%               IPv4 address for enp0s8: 192.168.29.1

* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroK8s to make it the smallest full K8s around.
  https://ubuntu.com/blog/microk8s-memory-optimisation

43 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Tue Mar 15 09:39:51 UTC 2022 on tty1
tom@vm-server:~$
```

Figure 12 : Interface d'accueil d'Ubuntu Server après la connexion

Ubuntu Server peut être un moyen simple et pratique de configurer un réseau domestique simple. Une autre fonctionnalité d'Ubuntu Server est la possibilité d'attribuer des tâches de « super utilisateur » pour faciliter l'administration du réseau, où l'utilisation de l'édition originale peut être plus difficile ou plus laborieuse.

II.2.2. Le RAID

Le RAID (Redundant Array of Independent Disk) est un ensemble de techniques inventé en 1987, qui permet de répartir le stockage de données sur plusieurs disques et en permettant aux actions d'écriture et de lecture de se chevaucher. En utilisant la redondance, une matrice RAID plus fiable que n'importe quel lecteur de disque.

Les matrices RAID apparaissent au système d'exploitation (OS) comme un seul disque dur logique. Le RAID utilise les techniques de mise en miroir de disque ou de striping de disque. La mise en miroir copie des données identiques sur plus d'un lecteur. Striping partitionne l'espace de stockage de chaque disque en unités

allant d'un secteur (512 octets) jusqu'à plusieurs mégaoctets. Les bandes de tous les disques sont entrelacées et adressées dans l'ordre.

La mise en miroir de disque et le striping de disque peuvent être combinés sur une matrice RAID. La mise en miroir et le striping sont utilisés ensemble dans les RAID 01 et RAID 10.

a) Le contrôleur RAID

Un contrôleur RAID peut être utilisé comme niveau d'abstraction entre le système d'exploitation et les disques physiques, présentant des groupes de disques comme des unités logiques. L'utilisation d'un contrôleur RAID peut améliorer les performances et aider à protéger les données en cas de crash.

Un contrôleur RAID peut être utilisé dans des baies RAID matérielles et logicielles. Dans un produit RAID matériel, un contrôleur physique gère la baie. Lorsqu'il se présente sous la forme d'une carte d'interconnexion de composants périphériques ou d'une carte PCI Express, le contrôleur peut être conçu pour prendre en charge des formats de disque tels que SATA et SCSI. Un contrôleur RAID physique peut également faire partie de la carte mère.

Avec le RAID logiciel, le contrôleur utilise les ressources du système matériel. Bien qu'il exécute les mêmes fonctions qu'un contrôleur RAID matériel, les contrôleurs RAID logiciel peuvent ne pas permettre d'augmenter autant les performances.

Si une implémentation RAID logicielle n'est pas compatible avec le processus de démarrage d'un système et que les contrôleurs RAID matériels sont trop coûteux, basés sur un firmware ou un pilote, RAID est une autre option d'implémentation.

Une puce de contrôleur RAID firmware est située sur la carte mère, et toutes les opérations sont effectuées par le CPU, comme pour le RAID logiciel. Cependant, avec le firmware, le système RAID n'est implémenté qu'au début du processus de démarrage. Une fois le système d'exploitation chargé, le pilote du contrôleur prend en charge la fonctionnalité RAID. Un contrôleur RAID firmware n'est pas aussi cher qu'une option matérielle, mais met plus de pression sur le CPU de l'ordinateur. Firmware-based RAID est également appelé Hardware-assisted software RAID, modèle hybride RAID et faux RAID.

b) Les types ou niveaux de RAID

Depuis l'invention du terme RAID, les auteurs ont distingué six niveaux de RAID, 0 à 5. Ce système numéroté leur a permis de différencier les versions et la façon dont elles utilisaient la redondance et répartissaient les données sur l'ensemble du réseau. Le nombre de niveaux a depuis lors augmenté et a été divisé en trois catégories : niveaux RAID standard, imbriqués et non standard.

Pour les besoins de notre recherche, nous étudierons les quatre niveaux les plus utilisés dans l'industrie.

- RAID 0 : Cette configuration a des rayures, mais aucune redondance des données. Il offre les meilleures performances, mais aucune tolérance aux pannes. Chaque fichier est repartitionné par petits morceaux sur plusieurs disques, à chaque opération (lecture, écriture) les disques sont sollicités ;

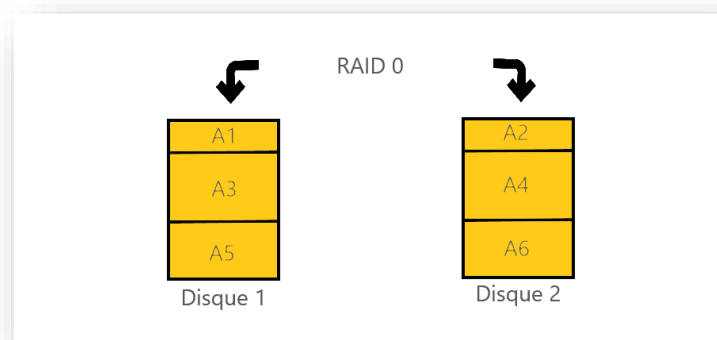


Figure 13 : Représentation du RAID 0

- RAID 1 (ou miroir) : se compose d'au moins deux lecteurs qui dupliquent le stockage des données. Il n'y a pas de rayures. Les performances de lecture sont améliorées puisque chaque disque peut être lu en même temps. Les performances d'écriture sont les mêmes que pour le stockage sur disque unique. Tous les disques sont des copies des autres ;

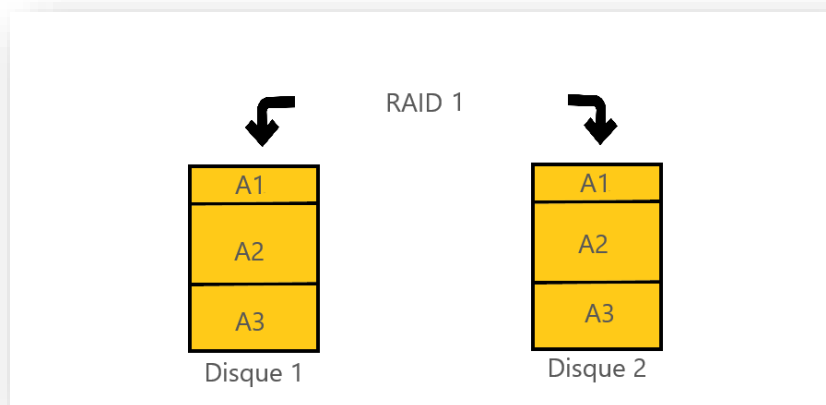


Figure 14 : Représentation du RAID 1

- RAID 10 (ou RAID 1 + 0) : Combinant RAID 1 et RAID 0, cette configuration offre des performances supérieures à RAID 1, mais à un coût beaucoup plus élevé. Dans RAID 1+0, les données sont mises en miroir et

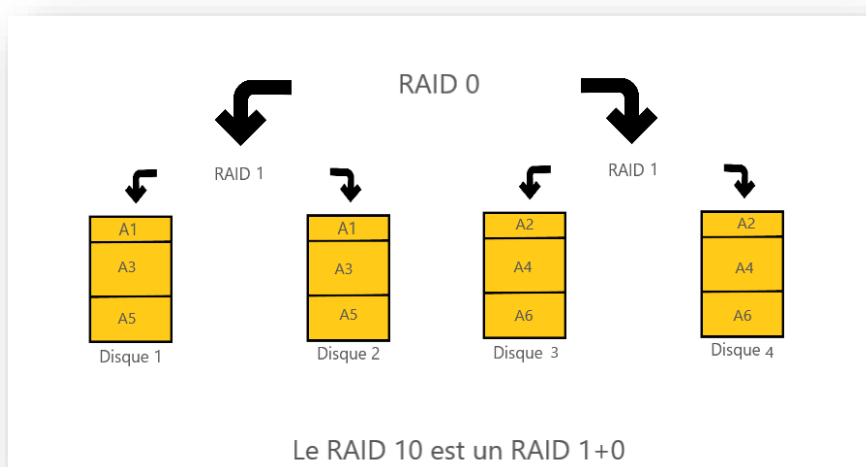


Figure 15 : Représentation du RAID 10

les miroirs sont rayés. Il regroupe les disques en grappes en RAID 1 et les fichiers sont réparties sur plusieurs disques en RAID 0 ;

- RAID 5 : Ce niveau est basé sur un striping de niveau bloc avec parité. L'information de parité est rayée sur chaque lecteur, ce qui permet au réseau de fonctionner même en cas de défaillance d'un lecteur. L'architecture du réseau permet des opérations de lecture et d'écriture sur plusieurs lecteurs. Il en résulte des performances généralement meilleures que celles d'un lecteur unique, mais pas aussi élevées que celles d'une baie RAID 0. RAID 5 nécessite au moins trois disques, mais il est souvent recommandé d'utiliser au moins cinq disques pour des raisons de performances.

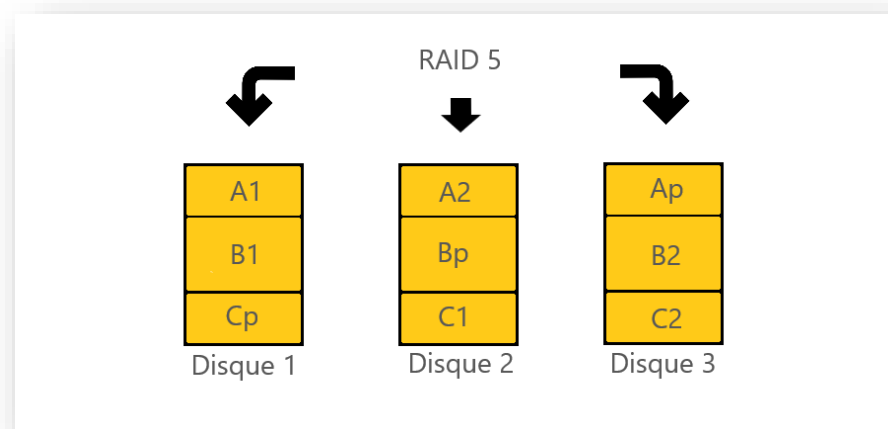


Figure 16 : Représentation du RAID 5

c) Avantages et Inconvénients

Chaque niveau de RAID présente un compromis performance/fiabilité/coût différent. Mais globalement, les avantages et inconvénients du RAID sont :

- ✓ Augmente la vitesse et la fiabilité des opérations ;
- ✓ Disponibilité accrue ;
- ✓ Coûts moindres ;
- ✓ Meilleure gestion des pannes ;

- ⊗ Reconstruction de disque en panne lourde ;
- ⊗ Risque élevé que plusieurs disques tombent en panne dans la même période ;
- ⊗ Peut être très coûteux ;
- ⊗ Risque de panne évolue en fonction de la capacité des disques.

II.2.3. LVM

LVM (ou Logical Volume Management) est un système de gestion de volumes logique basé sur trois niveaux d'abstraction : Physique, Groupe, Logique.

Il comprend l'allocation de disques, la répartition, la mise en miroir et le redimensionnement des volumes logiques. Avec LVM, un disque dur ou un ensemble de disques durs est alloué à un ou plusieurs volumes physiques. Les volumes physiques LVM peuvent être placés sur d'autres périphériques blocs qui peuvent s'étendre sur deux disques ou plus.

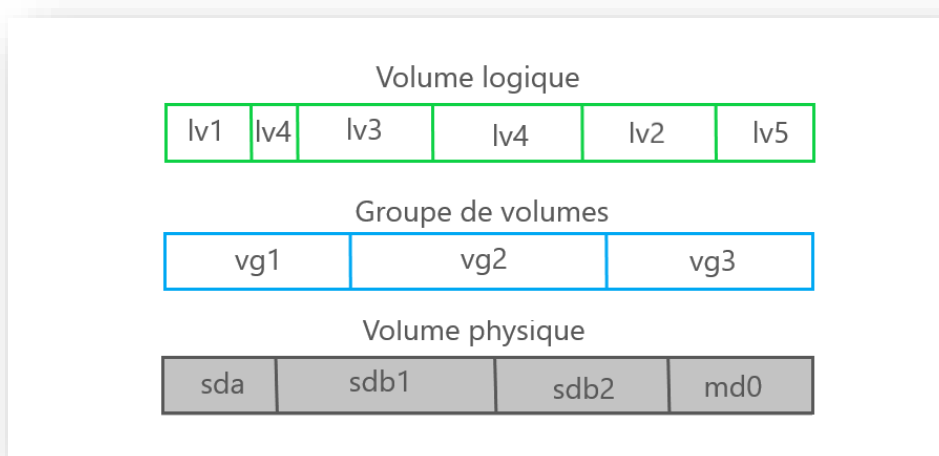


Figure 17 : Représentation des couches du modèle LVM

- La première couche concerne le volume physique. Un volume physique peut être un disque entier, une partition ou un volume RAID ;
- La seconde est constituée de groupe de volumes, c'est un regroupement des volumes physiques dans une boîte ou l'on peut ajouter ou retirer des volumes physiques ;
- La dernière couche concerne le volume logique. Les groupes de volumes peuvent être divisés en volumes logiques auxquels sont attribués des points de montage, des types de système de fichier (ext2, ext3).

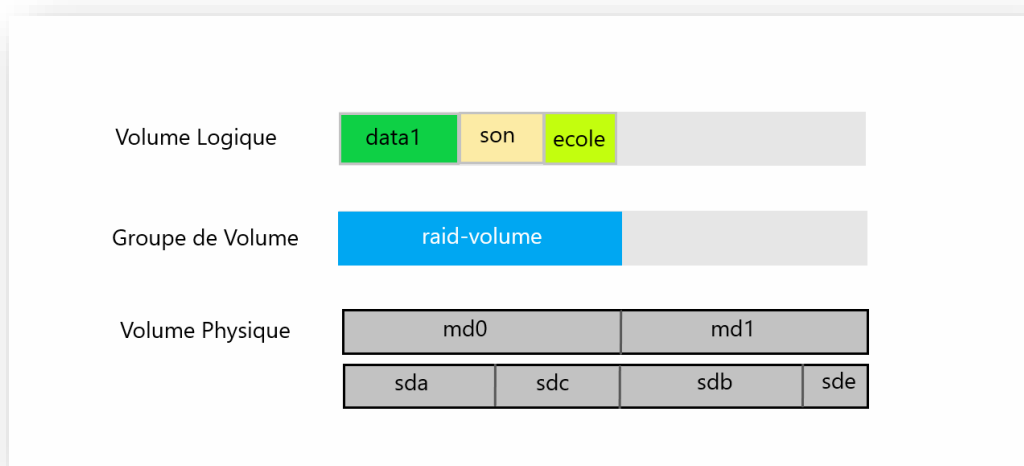


Figure 18 : Partitionnement des disques avec LVM

Lorsque les « partitions » atteignent leur pleine capacité, l'espace libre du groupe de volumes peut être ajouté au volume logique pour augmenter la taille de la partition. Lorsqu'un nouveau disque dur est ajouté au système, il peut être ajouté au groupe de volumes et la taille des partitions qui sont des volumes logiques peut être augmentée.

II.3. Architecture de programmation : MVC

C'est une architecture logicielle qui permet de sectionner le code en plusieurs parties logiques pouvant communiquer entre elles et rendre la programmation plus compréhensible et mieux maintenable.

a) Le Modèle : contient les données à afficher

C'est la partie de gestion des données. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve donc entre autres les requêtes d'accès à la base de donnée.

b) La Vue : contient la présentation de l'interface graphique

C'est la partie visuelle. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher.

c) Le Contrôleur : contient la logique concernant les actions effectuées par l'utilisateur

C'est la partie qui gère la logique et prend les décisions. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue.

II.4. Outils et Langages de programmation

Outil	Description	Logo
PHP	Langage de programmation libre permettant la production de pages Web dynamiques via un serveur http.	
Maria DB	Système de Gestion de Base de Données relationnelles basé sur le langage SQL.	
HTML	Langage de balisage conçu pour la représentation de pages web de base.	
CSS	Langage de feuille de style en cascade qui décrit la représentation des documents HTML.	
JavaScript	Langage de programmation de script coté client permettant des interactions entre l'utilisateur et les site web notamment via des Framework et des bibliothèques.	

Tableau 8 : Outils de développement

Section 3 : Conception du système

Pour rendre le futur système compréhensible pour la maîtrise d'œuvre et descellé les différents besoins quant à la réalisation du projet.

III.1 Diagramme de Cas d'Utilisation

III.1.1 Présentation

Un diagramme de cas d'utilisation est un type de diagramme utilisé en UML pendant la phase d'analyse d'un projet pour identifier la fonctionnalité du système. Il décrit l'interaction des personnes ou du dispositif externe avec le système en cours de conception. Il ne montre pas beaucoup de détails, mais résume seulement certaines des relations entre les cas d'utilisation, les acteurs et les systèmes.

Fondamentalement, quatre éléments doivent être inclus dans un diagramme de cas d'utilisation. Il s'agit des acteurs, des systèmes, des cas d'utilisation et des relations. Les acteurs représentent toute personne ou tout objet qui interagit avec le système. Il peut s'agir d'êtres humains, d'autres ordinateurs ou d'autres systèmes logiciels. Les cas d'utilisation représentent les actions qui sont effectuées par un ou plusieurs acteurs dans un but particulier. Le système est ce que vous êtes en train de développer.

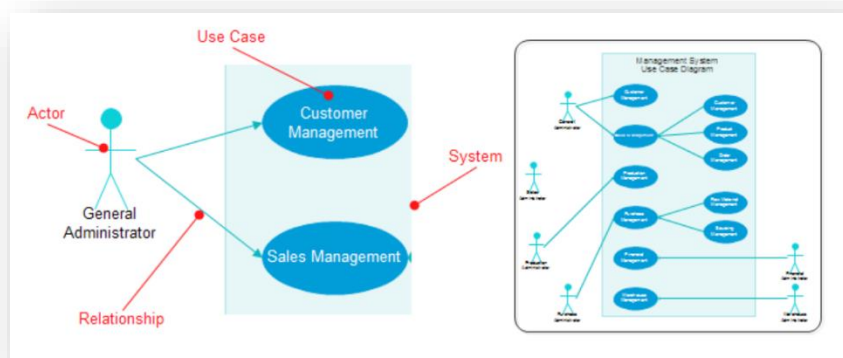


Figure 19 : Exemple de diagramme de cas d'utilisation [6]

III.1.2 Diagramme de Cas d'Utilisation du système

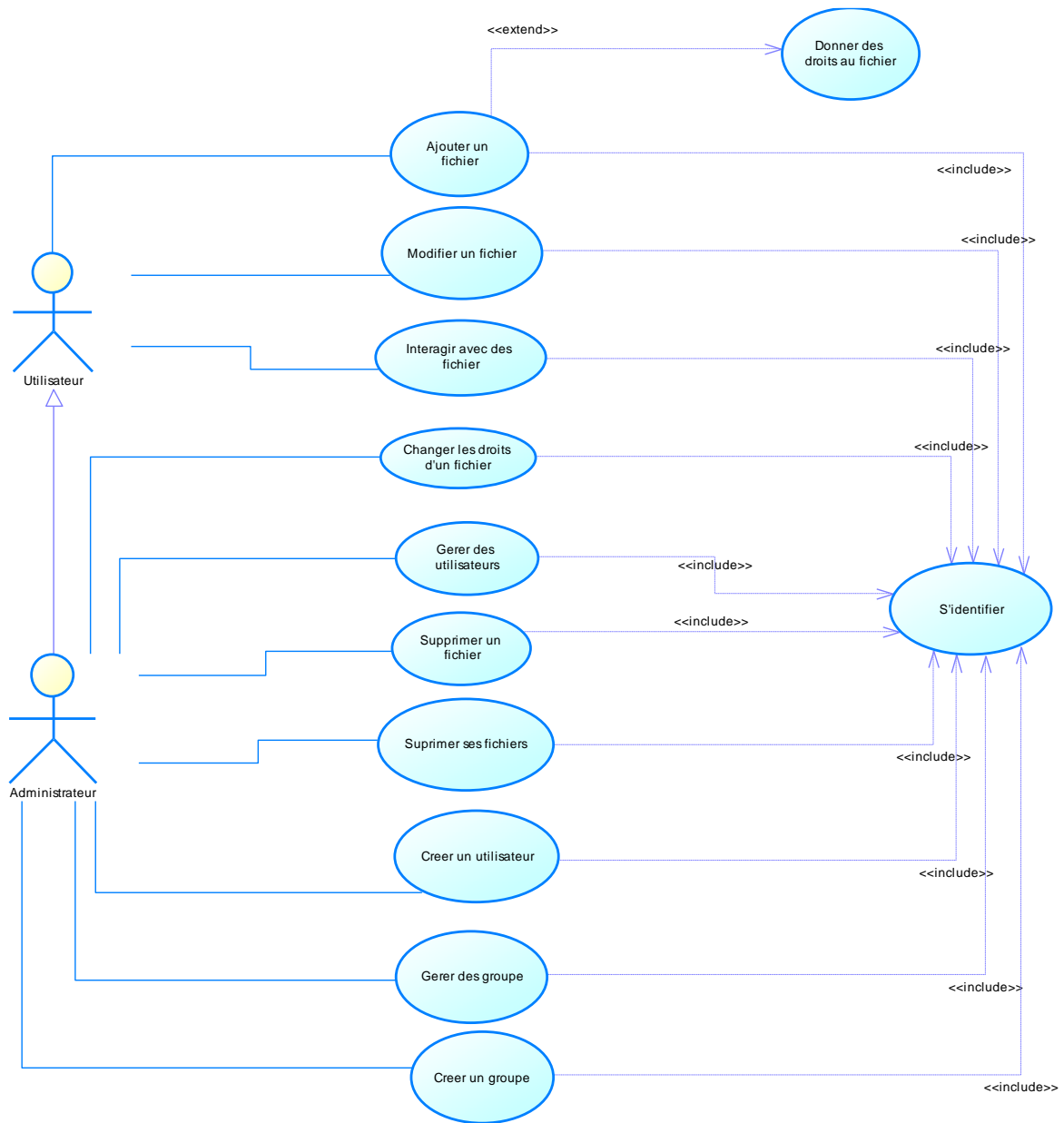


Figure 20 : Diagramme de cas d'utilisation

III.2 Diagramme de Séquence

III.2.1 Présentation

Un diagramme de séquence est utilisé en UML pour décrire les phases d'analyse et de conception. C'est un diagramme d'interaction qui détaille la manière dont les opérations sont effectuées. Un diagramme de séquence est souvent utilisé pour représenter le déroulement chronologique des événements à travers un cas d'utilisation. Il permet de présenter les relations de communication entre les objets et les messages qui déclenchent ces communications.

Une ligne de vie représente les instances typiques des composants ou des classes de votre système. Les messages sont affichés sous forme de flèches. Ils peuvent être complets, perdus ou trouvés ; synchrones ou asynchrones ; appel ou signal. Activer est utilisé pour indiquer l'activation d'un participant. Une fois qu'un participant est activé, sa ligne de vie apparaît. Les objets sont des éléments de modèle qui représentent des instances d'une ou plusieurs classes. Les classes en UML montrent l'architecture et les caractéristiques du système conçu. L'acteur spécifie un rôle joué par un utilisateur ou tout autre système qui interagit avec le sujet.

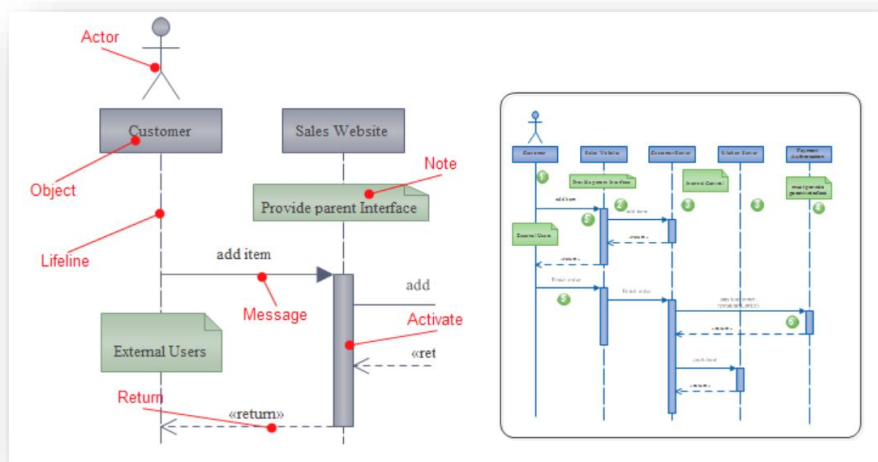


Figure 21 : Exemple de diagramme de séquence [6]

III.2.2 Diagramme de Séquence du système

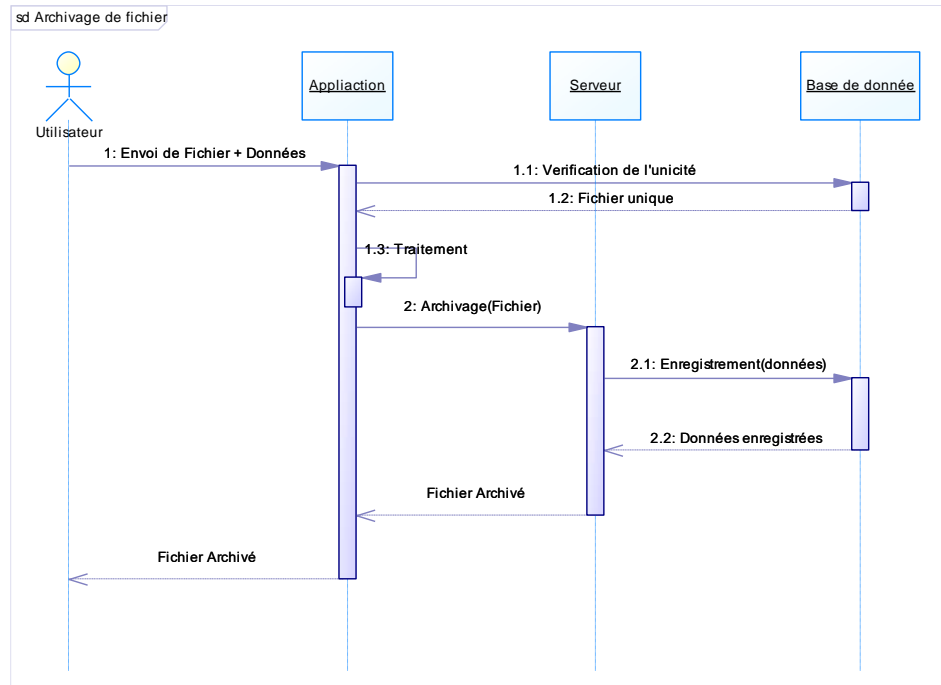


Figure 22 : Diagramme de séquence

III.3 Diagramme d'État-Transition

III.3.1 Présentation

Contrairement au diagramme d'activité qui aborde le système d'un point de vue global, le diagramme états-transitions cible un objet unique du système. Tous les

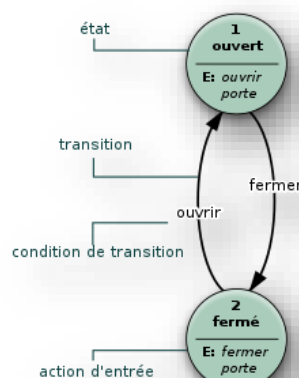


Figure 23 : Exemple de diagramme d'état transition [6]

automates du système s'exécutent parallèlement et peuvent donc changer d'état de façon indépendante.

III.3.2 Diagramme d'État-Transition du système

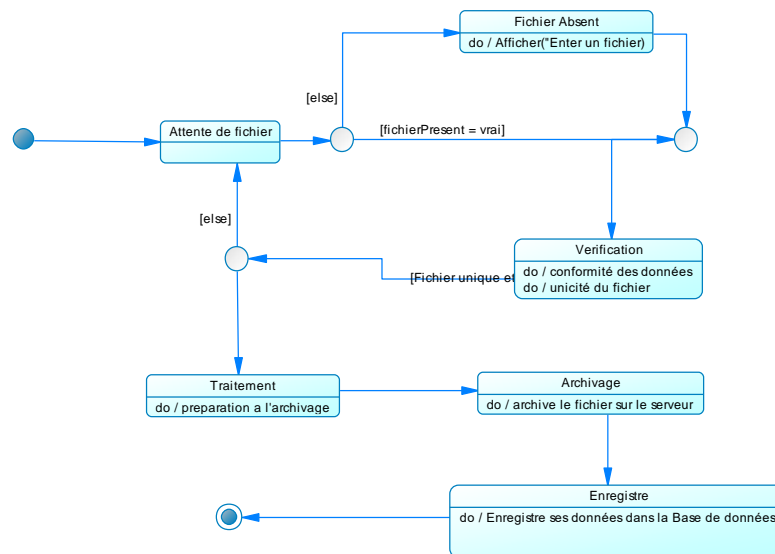


Figure 24 : Diagramme d'état transition

III.4 Diagramme d'Activité

III.4.1 Présentation

Le but d'un diagramme d'activité est de décrire le flux procédural des actions dans le cadre d'une activité. Il est utilisé pour modéliser la façon dont les activités sont coordonnées pour fournir un service, pour montrer les événements nécessaires à la réalisation d'une opération et pour illustrer les relations entre les événements dans un cas d'utilisation unique.

Les diagrammes d'activités sont constitués d'activités, d'états et de transitions entre les activités et les états. L'état initial est le point de départ du diagramme d'activités. C'est le point auquel vous commencez à lire la séquence d'action. Une activité est une unité de travail qui doit être exécutée. L'état définit

la condition actuelle d'un événement ou d'une activité. L'activité de décision est introduite en UML pour soutenir les conditions des activités. Elle montre où la transition de sortie d'un état ou d'une activité peut s'orienter dans d'autres directions en fonction d'une condition. La barre représente la synchronisation de l'achèvement de ces activités. Le flux de contrôle en informatique se réfère à l'ordre dans lequel les déclarations, instructions ou appels de fonction individuels d'un impératif ou d'un programme déclaratif sont exécutés ou évalués. Un flux d'objets est la même chose qu'un flux de contrôle, mais il est représenté par une ligne pointillée au lieu d'une ligne pleine.

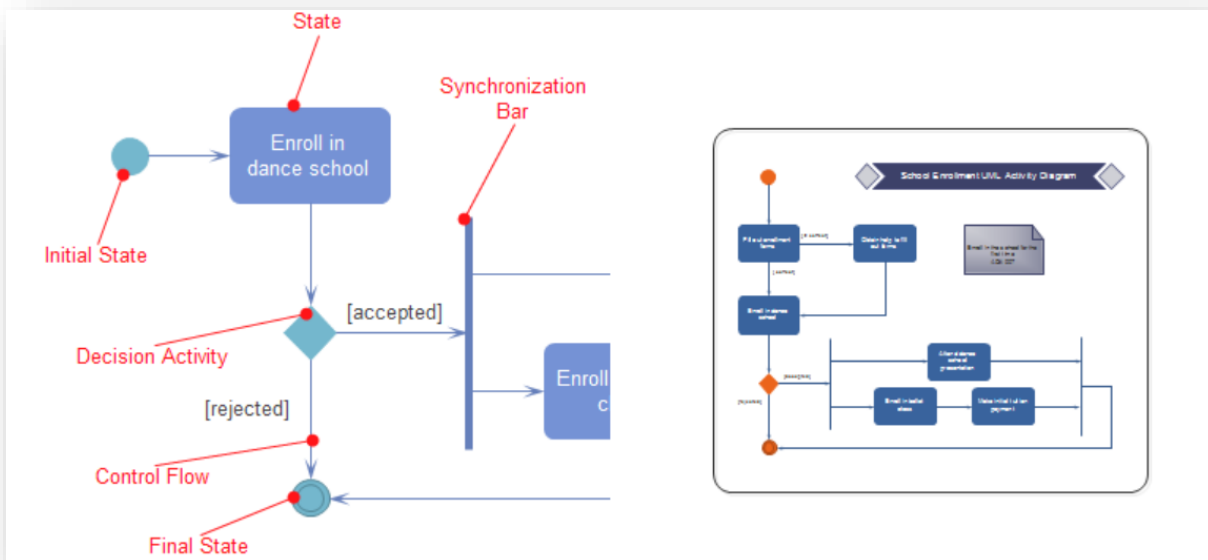


Figure 25 : Exemple de diagramme d'activité [6]

III.4.2. Diagramme d'Activité du système

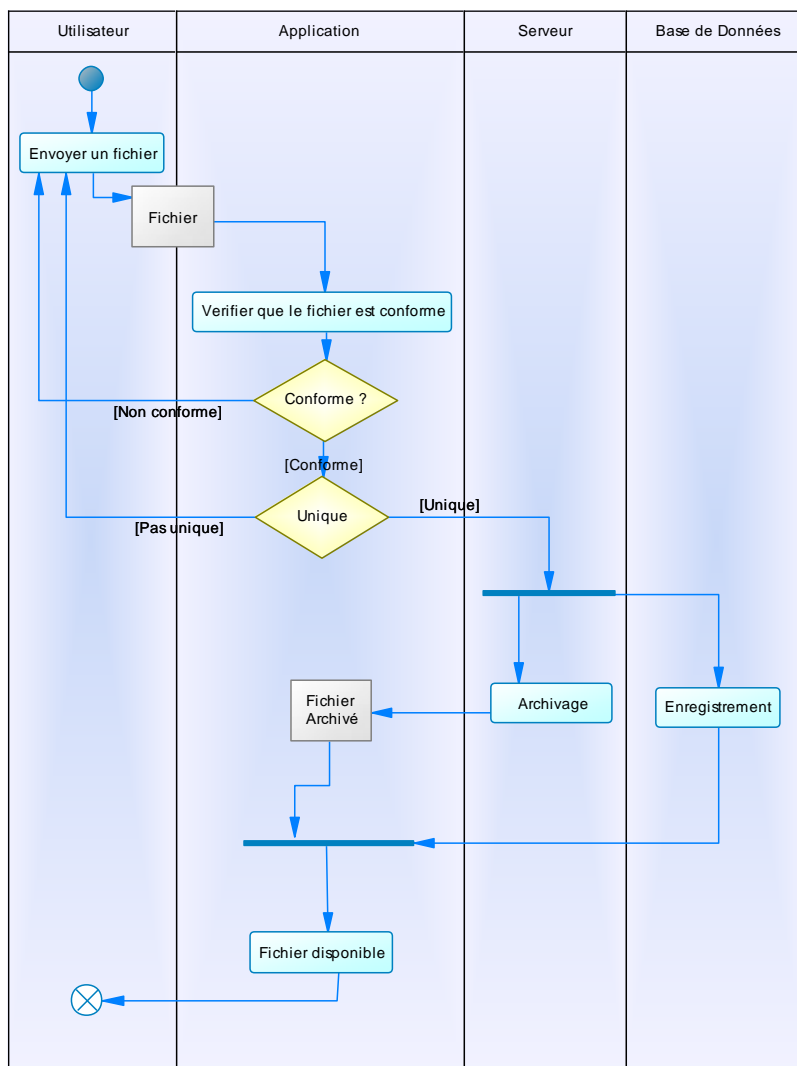


Figure 26 : Diagramme d'activité

III.5. Diagramme de Classe

III.5.1. Présentation

Un diagramme de classe UML est non seulement utilisé pour décrire les structures des objets et des informations dans une application, mais aussi pour montrer la communication avec ses utilisateurs. Il offre un large éventail d'utilisations, de la modélisation de la vue statique d'une application à la description des

responsabilités d'un système. La composition est un type spécial d'agrégation qui dénote une forte propriété.

Dans un diagramme de classes UML, les classes représentent une abstraction d'entités ayant des caractéristiques communes. Les associations représentent les relations statiques entre les classes. L'agrégation est un type spécial d'association dans lequel les objets sont assemblés ou configurés ensemble pour créer un objet plus complexe. La généralisation est une relation dans laquelle un élément de modèle (l'enfant) est basé sur un autre élément de modèle (le parent). La relation de dépendance est une relation dans laquelle un aspect, le client, utilise ou dépend d'un autre aspect, le fournisseur.

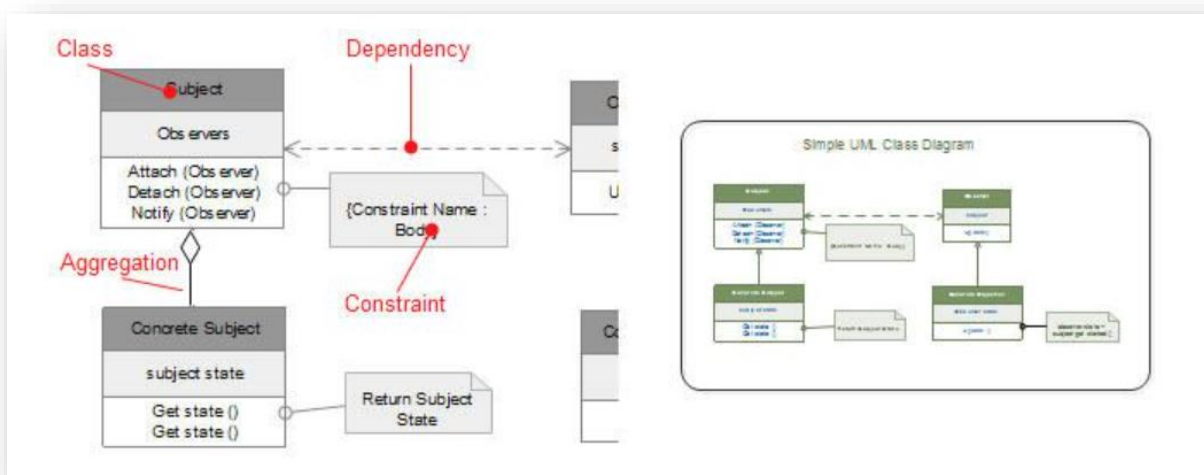


Figure 27 : Exemple de diagramme de classe [6]

III.5.2. Diagramme de Classe du système

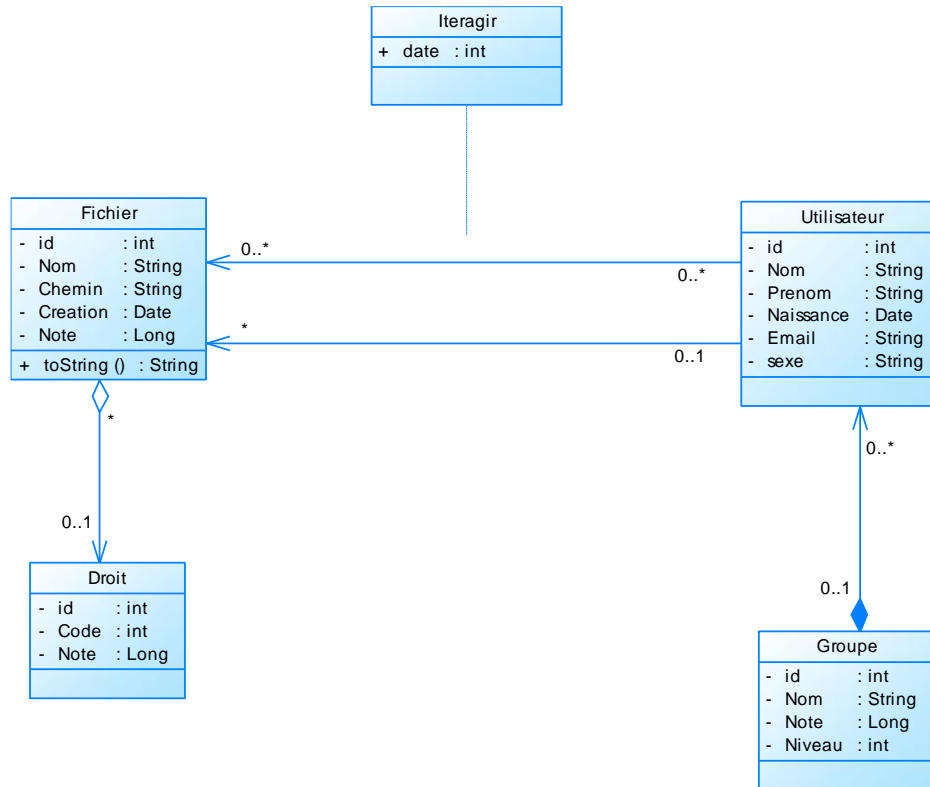


Figure 28 : Diagramme de classe

III.6. Diagramme d'Objet

III.6.1. Présentation

Le diagramme d'objet permet de représenter les instances des classes, c'est-à-dire des objets. Comme le diagramme de classes, il exprime les relations qui existent entre les objets, mais aussi l'état des objets, ce qui permet d'exprimer des contextes d'exécution. En ce sens, ce diagramme est moins général que le diagramme de classes. Les diagrammes d'objets s'utilisent pour montrer l'état des instances d'objet avant et après une interaction, autrement dit c'est une photographie à un instant précis des attributs et objets existants. Il est utilisé en phase exploratoire.

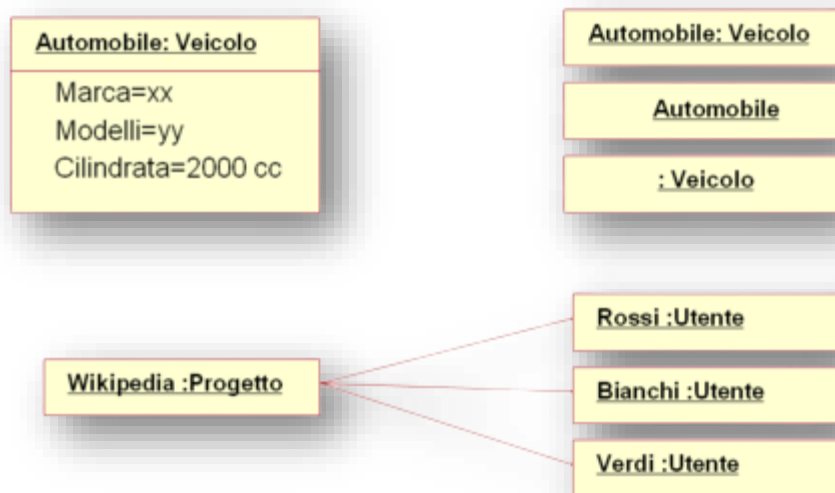


Figure 29 : Exemple de diagramme d'objet [6]

III.6.2. Diagramme d'Objet du système

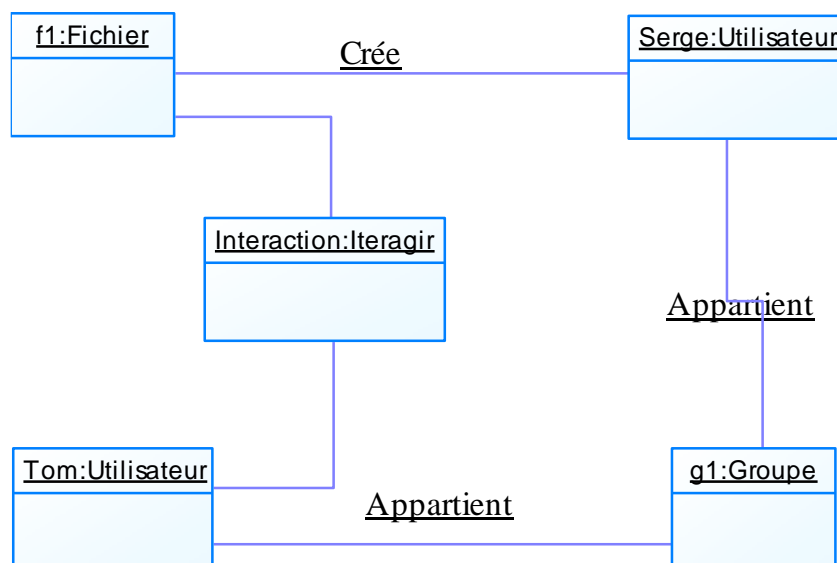


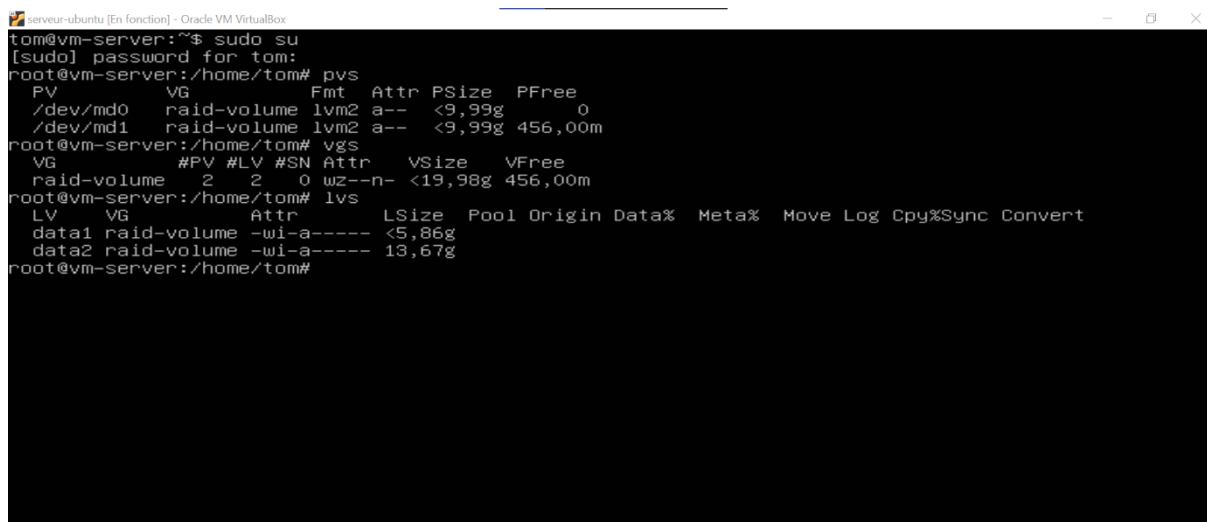
Figure 30 : Diagramme d'objet

Chapitre 4 : Résultat des tests

Dans ce chapitre nous exposerons les différents résultats de notre implémentation. Les résultats de la conception de notre système. Nous démontrerons par le biais de captures d'écran d'abord les résultats de la configuration de notre serveur (Section 1) et ensuite les interface de notre application (Section 2).

Section 1 : Configuration du serveur

I.1. Configuration des volumes RAID



```

serveur-ubuntu [En fonction] - Oracle VM VirtualBox
tom@vm-server:~$ sudo su
[sudo] password for tom:
root@vm-server:/home/tom# pvs
PV          VG          Fmt  Attr  PSize  PFree
/dev/md0    raid-volume  lvm2  a--   <9,99g  0
/dev/md1    raid-volume  lvm2  a--   <9,99g  456,00m
root@vm-server:/home/tom# vgs
VG          #PV #LV #SN Attr   VSize   VFree
raid-volume  2   2   0 wz--n- <19,98g 456,00m
root@vm-server:/home/tom# lvs
LV          VG          Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
data1       raid-volume -wi-a----- <5,86g
data2       raid-volume -wi-a----- 13,67g
root@vm-server:/home/tom#

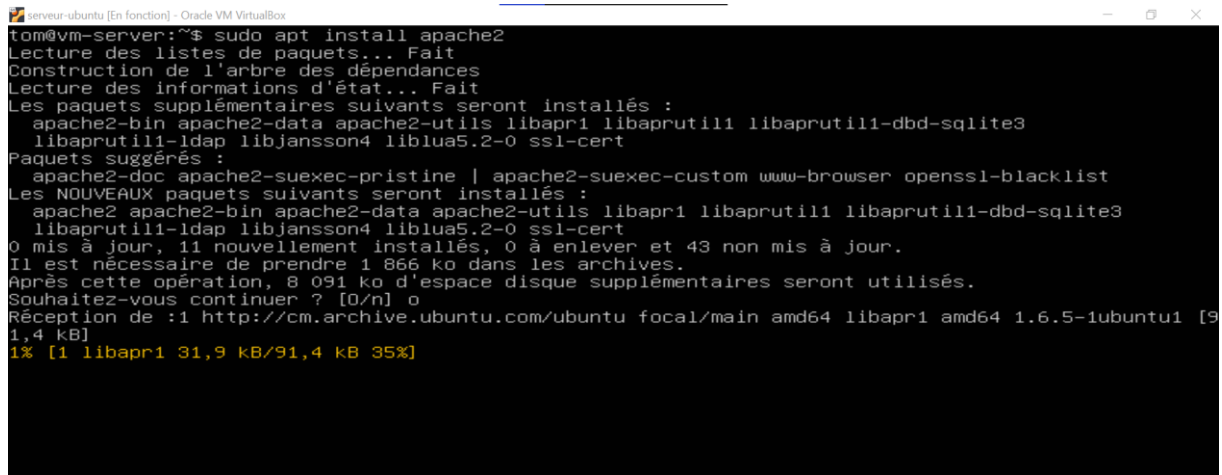
```

Figure 31 : Configuration des volumes RAID

Sur notre système Ubuntu Server, après avoir installé les différents paquets, et configuré le plan d'adressage, nous nous sommes attaqués à la gestion des disques.

Sur cette capture, nous voyons que nous avons deux disques durs physiques d'environ 10 giga octets faisant parti du même groupe de volume. Ce groupe de volume cumulant les deux volumes physiques a ensuite été découpé en deux volumes logiques nommés data1 d'environ 6 giga octets et data2 d'environ 14 giga octets.

I.2. Configuration du serveur WEB



```
tom@vm-server:~$ sudo apt install apache2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libjansson4 liblua5.2-0 ssl-cert
Paquets suggérés :
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser openssl-blacklist
Les NOUVEAUX paquets suivants seront installés :
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libjansson4 liblua5.2-0 ssl-cert
0 mis à jour, 11 nouvellement installés, 0 à enlever et 43 non mis à jour.
Il est nécessaire de prendre 1 866 ko dans les archives.
Après cette opération, 8 091 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
Réception de :1 http://cm.archive.ubuntu.com/ubuntu focal/main amd64 libapr1 amd64 1.6.5-1ubuntu1 [9
1,4 kB]
1% [1 libapr1 31,9 kB/91,4 kB 35%]
```

Figure 32 : Configuration du serveur WEB

La configuration du serveur web se complète à la planification d'adressage par l'installation et à la configuration d'outils. Ces outils nous permettront d'utiliser notre serveur comme un serveur web. Il s'agit ici des outils tels qu'Apache HTTP Server, MySQL, PHP pour ne citer que ceux-là.

Sur cette capture, cette commande nous a permis d'installer ce paquet depuis les dépôts d'Ubuntu.

La configuration d'apache se fait à l'aide du terminal et d'un éditeur tel que nano. Il s'agit de la configuration d'un Virtual host ou du localhost même, la fréquence de démarrage des services et les différents droits d'accès aux fichiers de configuration.

Section 2 : Programmes et Interfaces de Ralph Server

II.1. Programmes

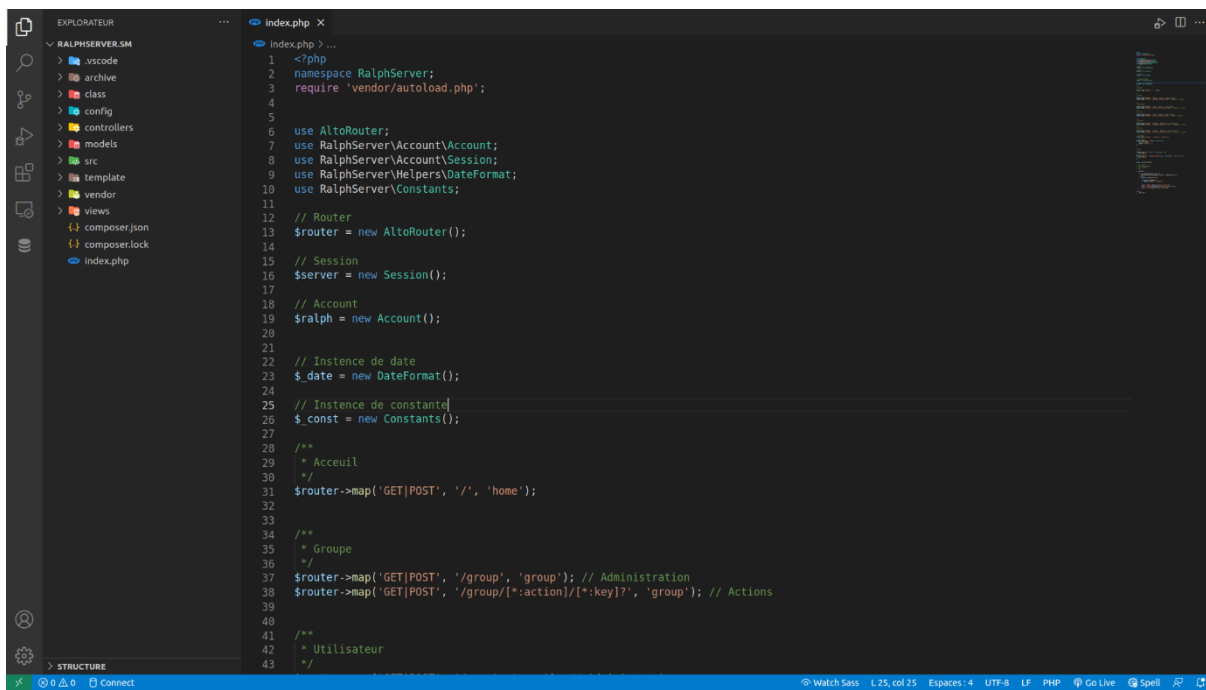


Figure 33 : Page d'index

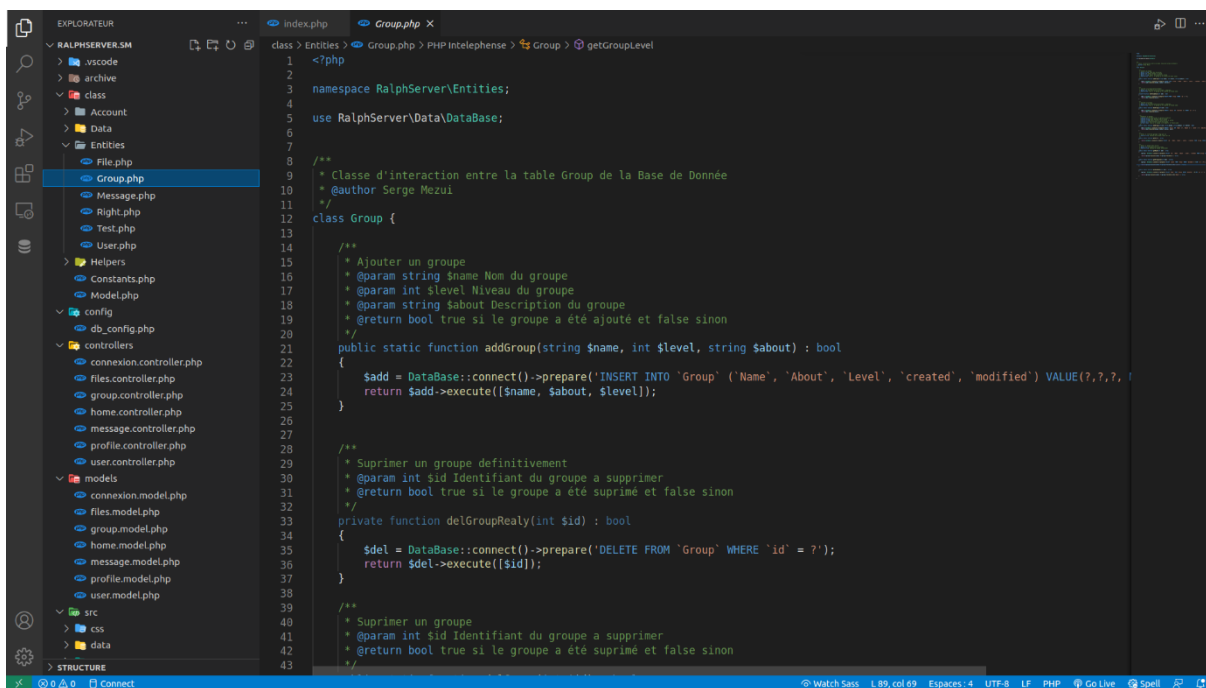


Figure 34 : Classe Groupe

Le code source de l'application a été faite en PHP selon l'architecture Model, Vue, Controller.

D'autres technologies tel que le HTML5, le CSS3 ou le JavaScript ont été utilisé pour réaliser ce projet.

L'architecture MVC a été soutenu par l'utilisation du paradigme orienté objet comme vous pouvez le constater sur cette capture, chaque entité du système a une classe qui renferme ses spécifications.

Les interfaces elles même ainsi que son design ont été faite à l'aide du Framework Bootstrap. Cet outil nous a permis de réaliser des interfaces plus conviviales. L'utilisation de quelques bibliothèques tierces du JavaScript nous ont permis de mieux dynamiser nos interactions avec l'utilisateur.

La base de données quant à elle a été faite en SQL à l'aide du SGBD MySQL. Elle nous a permis de mieux organiser les données avec des relations logiques et cohérentes.

II.2. Interfaces

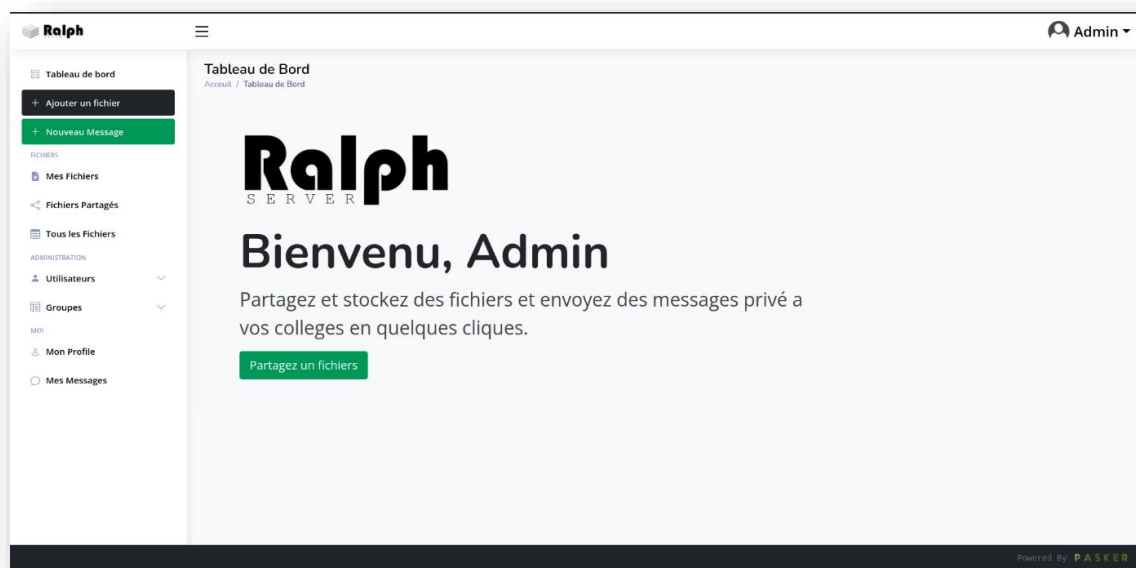


Figure 35 : Interface d'accueil d'administration

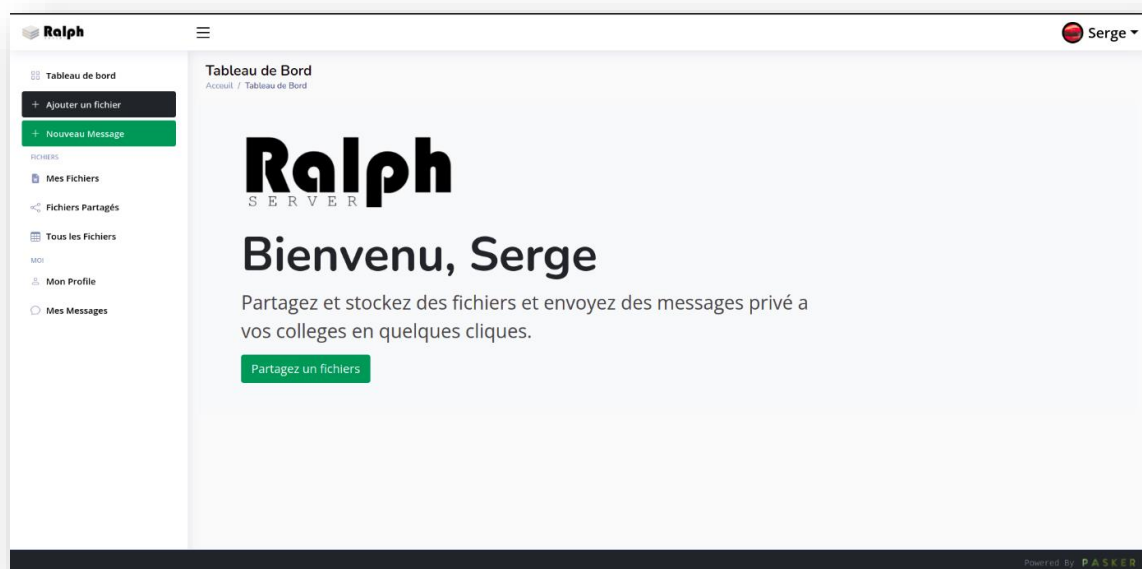


Figure 36 : Interface d'accueil d'utilisateur standard

Après la connexion à l'application, les interfaces d'accueil ci-dessus apparaitront en fonction de votre rôle dans le système.

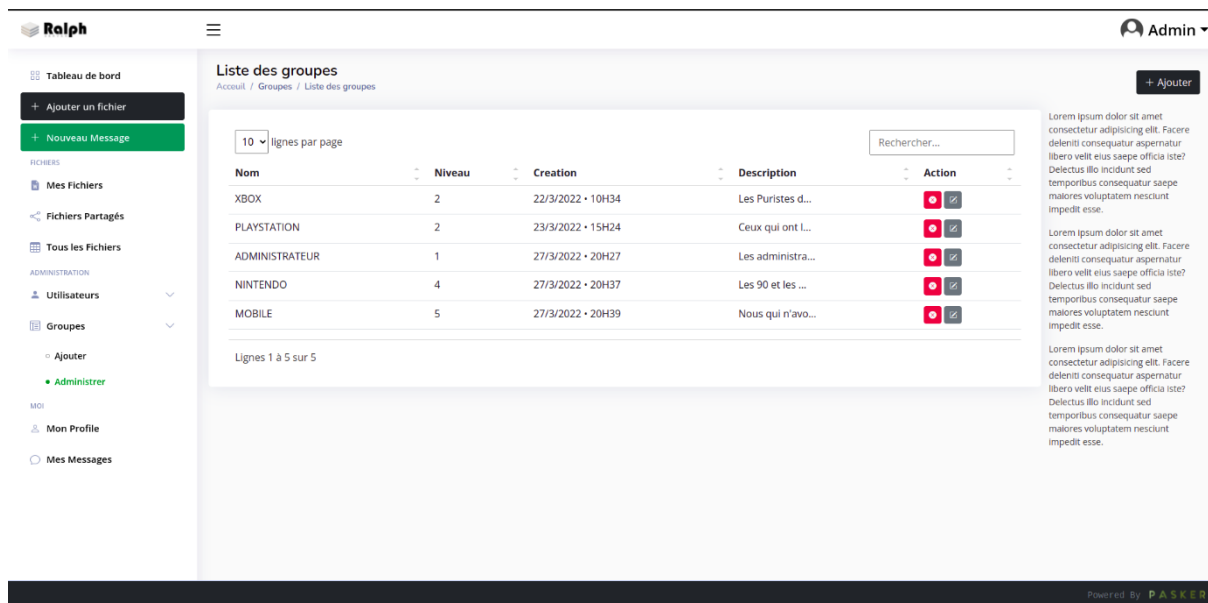


Figure 37 : Administration des groupes

Les administrateurs sont des utilisateurs qui ont certains droits élevés sur le système, comme par exemple l'ajout ou la suppression d'un utilisateur dans le système.

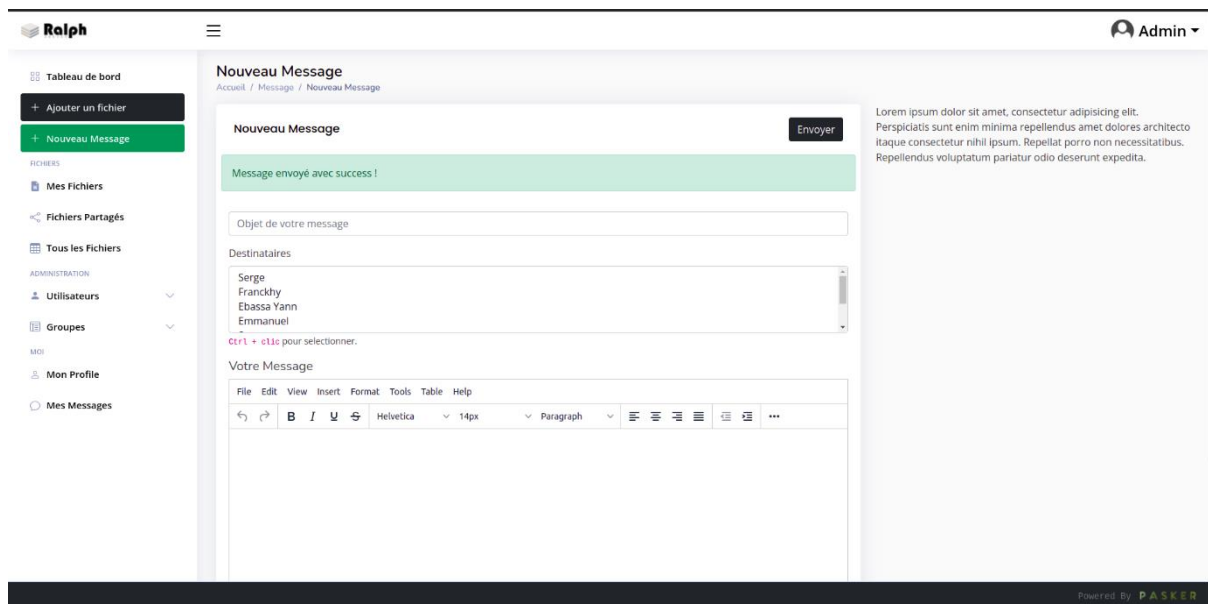


Figure 38 : Envoi de message privé

Tous les utilisateurs ont la possibilité de stocker et de partager de nouveaux fichiers sur le serveur. Ils peuvent aussi communiquer en envoyant des messages privés sous forme de mail.

CONCLUSION ET PERSPECTIVES

Le bilan de notre intervention au sein de l'entreprise e-Robot est positif. Cette excursion dans le monde professionnel nous a permis d'acquérir des connaissances et d'appliquer les enseignements reçus pendant ces deux dernières années de formation. Nous avons eu la chance de bénéficier d'un cadre professionnel propice à notre étude dans lequel nous avons pu pleinement exploiter les différentes ressources mis à notre disposition. Toutefois la mise en place d'un système d'un système de stockage et de partage de fichiers au sein de l'entreprise n'était pas aisée, mais la solution va permettre de faciliter la circulation des fichiers dans l'entreprise.

Le thème soumis à notre étude pour élaborer ce rapport, mise en place d'un système de stockage et de partage de fichiers, nous a permis de présenter la structure dans laquelle s'est déroulé notre stage, puis présentés les situations ayant mené à la conception de ce système et enfin à son implémentation via cette application.

La réalisation de ce projet a été pour nous un réel plaisir car elle nous a permis de nous lancer pleinement dans la réalisation d'un projet au vu de résoudre un problème commun à plusieurs entreprises. Elle nous a permis de mieux comprendre ce pourquoi nous étions formés.

Dans le but d'améliorer notre système, nous souhaitons que notre application puisse permettre de digitaliser plusieurs tâches manuelles en entreprise. Une amélioration serait d'implémenter un système de téléphonie intégré à notre application et un système de messagerie instantané.

RÉFÉRENCES BIBLIOGRAPHIQUE

Bibliographie

- [1] Pierre-Alain Goupille, *Technologie des ordinateurs et des réseaux* 8^e Edition.
- [2] Gerard Delamarre, *Dictionnaire des réseaux télématique* RVA EDI Collection TRANSPAC.

Webographie

- [3] Red Hat, *Le Stockage Réseau c'est quoi ?* <https://www.redhat.com/fr/topics/data-storage/network-attached-storage> ; Consulté le 12/02/2022.
- [4] Red Hat, *Stockage en mode fichier, bloc ou objet ?* <https://www.redhat.com/fr/topics/data-storage/file-block-object-storage> ; Consulté le 12/02/2022.
- [5] Techopedia, *Ubuntu Server*. <https://www.techopedia.com/definition/4207/ubuntu-server> ; Consulté le 22/02/2022.
- [6] Edrawsoft, *Les diagrammes UML*. <https://www.edrawsoft.com/fr/uml-introduction.html> ; Consulté le 13/03/2022.

ANNEXE

Logo de l'application Ralph Server



Figure 39 : Logo Ralph Server

TABLE DES MATIERES

<i>DEDICACE</i>	<i>i</i>
<i>REMERCIEMENTS</i>	<i>ii</i>
<i>RÉSUMÉ</i>	<i>iii</i>
<i>ABSTRACT</i>	<i>iv</i>
<i>SOMMAIRE</i>	<i>v</i>
<i>LISTE DES FIGURES</i>	<i>vi</i>
<i>LISTE DES TABLEAUX</i>	<i>vii</i>
<i>LISTE DES ABRÉVIATIONS</i>	<i>viii</i>
<i>INTRODUCTION GÉNÉRALE</i>	<i>1</i>
<i>Chapitre 1 : Présentation Générale de l'entreprise</i>	<i>3</i>
Section I : Présentation de l'entreprise e-Robot	4
I.1. Fiche Signalétique	4
I.2. Partenaires	5
I.3. Organigramme	5
Section 2 : Accueil dans l'entreprise	6
II.1. Présentation et Déroulement du stage	6
II.2. Plan d'affectation	6
II.3. Chronologie	6
II.3.1. Première affectation : Département de Design	6
II.3.2. Seconde affectation : Département de Développement	7
II.3.3. Troisième affectation : Département des ressources Humaines	7
II.2. Taches effectuées	8
II.4. Difficultés rencontrées	8
II.5. Bénéfices	8
<i>Chapitre 2 : Description du projet</i>	<i>9</i>
Section 1 : Contexte et Problématique	10
I.1. Contexte	10
I.2. Problématique	10
Section 2 : Étude de l'existant	11
II.1. État de l'art	11

II.1.1. Le stockage de masse ou stockage physique	11
II.1.2 Le stockage Réseau	12
II.1.2.1 Le NAS	13
a) Matériels et logiciels	13
b) Protocol	14
c) Avantages	15
II.1.2.2 Le SAN	15
a) Matériels et logiciels	15
b) Avantages	16
II.1.3 Le stockage Cloud	17
II.2 Étude de l'existant	18
II.2.1 Description du système de gestion des documents	18
a) Gestion des documents physiques	18
b) Gestion des documents numériques	18
c) Partage des fichiers dans l'entreprise	18
II.3 Critique de l'existant	19
II.3.1 Avantages	19
II.3.2 Inconvénients	19
Section 3 : Situation ayant conduit à ce travail	20
III.1 Description	20
III.2 Proposition de solution	20
Section 4 : Cahier des charges	21
IV.1 Contexte et justifications	21
IV.1.1 Contexte	21
IV.1.2 Problème	21
IV.2 Objectifs	21
IV.2.1 Objectif globale	21
IV.2.2 Objectif spécifique	22
IV.3 Besoins	22
IV.3.1 Besoins fonctionnels	22
IV.3.2 Besoins non fonctionnels	22
IV.4 Livrables	23
IV.5 Planification et estimation des coups	23
IV.5.1 Planification du projet	23
IV.5.2. Estimation des couts	24
a) Ressources logicielles :	25
b) Ressources matérielles :	26
c) Ressources humaines	26
d) Autres ressources	27

e) Récapitulatif	27
<i>Chapitre 3 : Méthodologie et technique</i>	28
Section 1 : Méthode de conception	29
I.1. Processus Unifié ou UP	29
a) Méthodologie	29
b) Objectifs	30
c) Rôle	31
i) Expression des besoins :	31
ii) L'analyse	31
iii) Conception	32
I.2. Le Langage UML	32
Section 2 : Outils de conception	35
I.1. Outils	35
II.1.1. Power AMC	35
II.1.2. GANTT Project	36
II.2. Réseau et Administration système	37
II.2.1. Ubuntu Server	37
II.2.2. Le RAID	38
a) Le contrôleur RAID	39
b) Les types ou niveaux de RAID	40
c) Avantages et Inconvénients	42
II.2.3. LVM	43
II.3. Architecture de programmation : MVC	45
a) Le Modèle : contient les données à afficher	45
b) La Vue : contient la présentation de l'interface graphique	45
c) Le Contrôleur : contient la logique concernant les actions effectuées par l'utilisateur	45
II.4. Outils et Langages de programmation	46
Section 3 : Conception du système	47
III.1 Diagramme de Cas d'Utilisation	47
III.1.1 Présentation	47
III.1.2 Diagramme de Cas d'Utilisation du système	48
III.2 Diagramme de Séquence	49
III.2.1 Présentation	49
III.2.2 Diagramme de Séquence du système	50
III.3 Diagramme d'État-Transition	50
III.3.1 Présentation	50
III.3.2 Diagramme d'État-Transition du système	51
III.4 Diagramme d'Activité	51



III.4.1 Présentation	51
III.4.2. Diagramme d'Activité du système	53
III.5. Diagramme de Classe	53
III.5.1. Présentation	53
III.5.2. Diagramme de Classe du système	55
III.6. Diagramme d'Objet	55
III.6.1. Présentation	55
III.6.2. Diagramme d'Objet du système	56
<i>Chapitre 4 : Résultat des tests</i>	57
Section 1 : Configuration du serveur	58
I.1. Configuration des volumes RAID	58
I.2. Configuration du serveur WEB	59
Section 2 : Programmes et Interfaces de Ralph Server	60
II.1. Programmes	60
II.2. Interfaces	62
<i>CONCLUSION ET PERSPECTIVES</i>	64
<i>RÉFÉRENCES BIBLIOGRAPHIQUE</i>	65
Bibliographie	65
Webographie	65
<i>ANNEXE</i>	66
Logo de l'application Ralph Server	66
<i>TABLE DES MATIERES</i>	67