PROJET INTERDISCIPLINAIRE

**ISIMs Park – Location de vélos**

Année Académique 2023-2024

**Groupe N°2**

Thibault Herry, Constantinos Samaras,

Mattias Bruneau, Arnaud De Lena, Fabio Mirasola

### Table d’illustrations

Figure 1 :Plan d'adressage 6

Figure 2 : Cycle en V 8

Figure 3 : MLD 11

Figure 4 : MCD 11

Figure 5 : AD + DNS 15

Figure 6 : DHCP 15

Figure 7 : BackUp 16

### Table des matières

[Table d’illustrations 2](#_Toc154154127)

[Table des matières 3](#_Toc154154128)

[1. Introduction 4](#_Toc154154129)

[2. Présentation du groupe 5](#_Toc154154130)

[ Répartition des tâches 5](#_Toc154154131)

[3. Présentation générale du projet 6](#_Toc154154132)

[4. Documents utiles 7](#_Toc154154133)

[ QQOQCP 7](#_Toc154154134)

[ Cycle en V 8](#_Toc154154135)

[5. Présentation de la partie programmation 9](#_Toc154154136)

[ Front-End 9](#_Toc154154137)

[ Back-End 10](#_Toc154154138)

[6. Présentation de la DB 11](#_Toc154154139)

[ MLD/ MCD 11](#_Toc154154140)

[ Base de données 12](#_Toc154154141)

[7. Présentation de la partie Linux Server 13](#_Toc154154142)

[ Partitionnement 13](#_Toc154154143)

[ Les rôles : 13](#_Toc154154144)

[ Packages Installés 14](#_Toc154154145)

[ Sécurité 14](#_Toc154154146)

[ Backups 14](#_Toc154154147)

[8. Présentation de la partie Windows Server 15](#_Toc154154148)

[9. Problèmes rencontrés 17](#_Toc154154149)

[10. Conclusion 18](#_Toc154154150)

[11. Bibliographie 19](#_Toc154154151)

# Introduction

Dans le cadre de notre cours « Projet interdisciplinaire », il nous a été demandé de réaliser un projet en suivant un cahier des charges donné.   
Ce projet consiste à créer différents modules web afin d’implémenter un service de gestion de locations / réparations de vélos. Nous expliquerons le projet en détails plus tard dans le rapport. Par ailleurs, nous devions également nous occuper de la mise en place de l’infrastructure réseau (installation de serveur DNS, DHCP, gestion d’un AD, Linux).

Pour effectuer ces différentes tâches, nous étions réunis par groupe de 5 étudiants choisis aléatoirement par les professeurs. Il était important pour nous de répartir judicieusement les tâches selon les compétences de chacun.   
Pour réaliser ce projet, nous disposions de 5 jours complet de cours, du lundi au vendredi de 8h15 à 17h30 afin de tout réaliser.

Dans un premier temps, notre premier réflexe a été de bien analyser en groupe le cahier des charges fournis par le client. Ainsi, dès que nous nous sommes assuré de la bonne compréhension de ce dernier, nous avons commencé à créer les différents outils afin de bien démarrer le projet. Nous avons réalisé un QQOQCP ainsi qu’un diagramme de cycle en V. Ces derniers nous ont permis d’avoir une vue d’ensemble de notre projet et ainsi pouvoir nous diriger correctement sur le plan organisationnel.

Par la suite, nous nous sommes concentrés sur la réalisation de l’infrastructure réseau. Nous avons réparti les tâches entre les 2 gestionnaires systèmes. Mattias s’est occupé de l’installation et la configuration de Windows Server et Thibault s’est occupé de celle du serveur web sur linux. Ces dernières seront expliquées plus amplement dans les points 7 et 8 du rapport.

Pendant ce même temps, les développeurs se préoccupaient de déterminer les modules ou zones de développement nécessaires. Ce projet étant assez conséquent, il était important de s’organiser et de développer tous les modules en parallèle. Nous avons commencé par élaborer les schémas de structure de la base de données (MLD/MLC). Nous avons ensuite créé des wireframes que nous avons présentés au client.

# Présentation du groupe

Chef d’équipe : HERRY Thibault  
Développeurs : SAMARAS Constantinos, DE LENA Arnaud, MIRASOLA Fabio. Gestionnaires systèmes : BRUNEAU Mattias, HERRY Thibault

## Répartition des tâches

* Bruneau Mattias : Gestion de Windows Server (création & gestion de l’AD, mise en place de serveur DNS, DHCP, système de backup), ajout du CSS, rapport
* Herry Thibault: Gestion de l’administration Linux, CSS, rapport
* Samaras Constantinos : Implémentation du code, la partie système, rapport
* De Lena Arnaud : Implémentation du code, partie système, rapport
* Mirasola Fabio : implémentation du code, diagrammes, guide utilisateur, rapport.

# Présentation générale du projet

Notre projet consiste en la création de différents modules web afin de gérer un service de locations / réparations de vélos pour des utilisateurs. Ces derniers auront la possibilité de réserver un ou plusieurs vélo(s) selon leurs envies. Il existe également plusieurs types de vélos : électrique, ville, VTT, vélo avec chariot pour enfants. Ils pourront récupérer leur(s) vélo(s) à l’aide de leur code QR délivré sur le site internet. Grâce à ce même code, ils auront accès à leurs réservations, ils pourront également prendre rendez-vous afin d’emmener leur vélo en réparation. Un calendrier sera disponible sur le site pour obtenir une vue d’ensemble de leur réservation / réparation.

Notre projet dispose aussi d’une partie Administrateur, ces derniers pourront avoir une vue d’ensemble et gérer la gestion du stock de vélos comprenant ; l’ajout, la suppression, l’édition et enfin la réparation des vélos. Ils pourront par ailleurs gérer le planning des clients et aussi récupérer les vélos loués.

Faisons le point au niveau infrastructure réseau. Nous avions certains besoins afin de mener au mieux la configuration de notre réseau. Dans un premier temps, nous avons créé notre serveur Windows avec les rôles Active Directory, DNS et enfin DHCP. Ce dernier sera expliqué dans son point respectif.

Nous avons également installé un serveur web hébergé sur une machine virtuelle fonctionnant sous AlmaLinux.

Une image contenant diagramme, ligne, cercle, carte

Description générée automatiquementCi-dessous un plan d’adressage du réseau.

Figure 1 :Plan d'adressage

# Documents utiles

## QQOQCP

|  |  |  |
| --- | --- | --- |
|  | QQOQCCP | |
| □ | QUOI | Site web permettant la réservation, la réparation ainsi que la gestion de locations de vélos dans l’Isims Park |
| □ | QUI | Le principal utilisateur de notre site sera le client de l’Isims Park mais aussi les employés comme l’administrateur qui aura des permissions pour gérer les réservations ainsi que les demandes de réparations.  Nous sommes trois developpeurs et deux administrateurs réseaux sur ce projet. |
| □ | OU | Le client aura accès à cette page en tout temps mais il devra se connecter avant de faire une demande. La demande d’une réservation d’un vélo est réalisable seulement si le client a réservé un logement.  Toutes les informations concernant la réservation ou la réparation d’un vélo se trouveront sur le site. |
| □ | QUAND | Le client utilisera le service quand il aura réserver au préalable un séjour dans le par cet/où quand il y sera déjà. Il pourra après sa réservation du logement, réserver un crénau pour louer un vélo et/ou réaliser une réparation s’il est déjà dans le complexe. |
| □ | COMMENT | Le client utilisera le produit afin de réserver un créneau horaire(en jour) pour obtenir un vélo. Aussi il pourra demander un rendez-vous vis-à-vis des heures disponibles afin de réparer un vélo défectueux. Tout ça en étant client avec une réservation valable de son logement. |
| □ | COMBIEN | Le prix de la location du server, les salaires des trois développeurs ainsi que des deux administrateurs réseaux.  Le projet sera réalisé en 5 jours à hauteur de 200€ par jour et par personne. |
| □ | POURQUOI | Nous réalisons ce projet dans le but d’automatiser la gestion des vélos afin de faire gagner du temps aux employés ainsi qu’aux clients du complexe. |
|  | NEXT STEP | Cycle en V |

## Cycle en V

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

Figure 2 : Cycle en V

# Présentation de la partie programmation

## Front-End

Nous avons d’abord créé les Wireframes selon les demandes du client. Afin de les réaliser, nous avons utilisé le logiciel Goodnotes. Concernant les layouts, ils ont été réalisés avec le logiciel Figma. Pour les images, nous avons cherché des icones libres de droit. Les couleurs de fond, nous avons utilisé un code CSS sur « CSSGradient ».

Premièrement, nous avons implémenté le code de la nav bar afin qu’elle serve de modèle pour toutes les pages, évitant ainsi la répétition de code dans les différentes vues. Ensuite nous avons créé un gabarit, celui-ci contient la partie HTML de base avec le contenu de la head qui contient les différentes balises « meta » ainsi que la partie « body » qui elle-même contient une route pointant vers la nav bar mais également vers chaque view différente suivant l’endroit où on se situe.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquementDeuxièmement, nous avons réalisé différentes view suivant le nombre de page que notre site contiendra. Etant donné que nous avons réalisé un code « template », chaque view que l’on aura implémenté sera relié à celui-ci afin d’afficher la view nécessaire. Par exemple lorsque l’on sera sur la page « login », la view « login » sera automatiquement générée par le fichier « template ».

## Back-End

Avant d’entamer la programmation, nous nous sommes occupés de la création de la base de données. Pour ce faire, nous avons réalisé le MLD et MCD suivant les fonctionnalités et les demandes que le client nous a fournis avec le cahier des charges. Une fois la modélisation mise au point, nous avons créé la base de données sur phpMyAdmin en suivant nos schémas. Nous avons exporté la base de données en tant que « isimsparkg2.sql »

Ensuite place au code, nous avons suivis les principes de l’architecture MVC, et de la POO dès le départ, pour garder un haut niveau de modularité du code afin de pouvoir développer de manière efficace et de permettre le développement en parallèle.

Il y a également la partie « core » qui représente le « squelette » de l’application, elle reprend les différentes classes de base nécessaires au fonctionnement du code, comme le contrôleur de base à implémenter, le fichier « configuration » qui aura la tâche de charger la configuration du programme, une classe « model » qui implémente les opérations de base d’un modèle, la classe « view » qui gère les opérations nécessaires à l’affichage des vues, la classe « route » qui se chargera de lier une URI à un contrôleur. Et finalement, la classe « routeur », qui elle s’occupera d’exécuter la méthode du contrôleur correspondant à la route demandée.

Nous avons aussi créé différentes classes contenant des fonctions générales afin de pouvoir les réutiliser dans les différents fichiers notamment pour la gestion d’erreurs, la gestion de la connexion LDAP, la vérification des privilèges, la gestion de variables de session.

Le nom du fichier de configuration est défini dans le fichier « config », cela servira donc à pouvoir changer le fichier de configuration sans modifier directement le code.

# Présentation de la DB

## MLD/ MCD

Afin de réaliser la base de données, nous avons tout d’abord réfléchi aux différentes tables à réaliser suivant les modèles MCD et MLD.

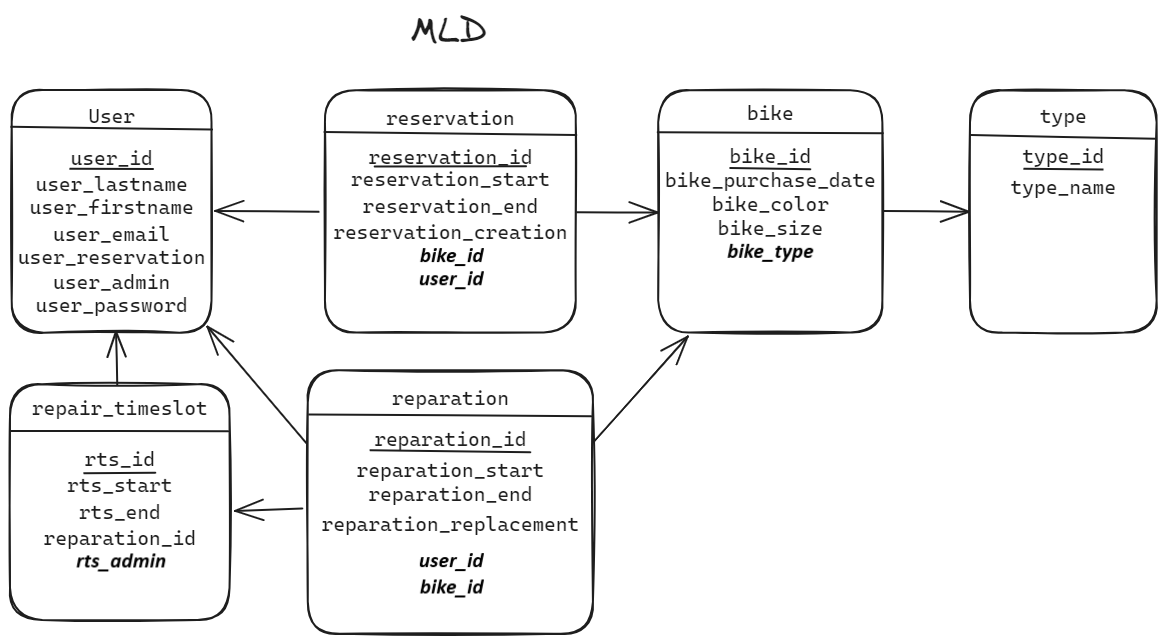


Figure 3 : MLD

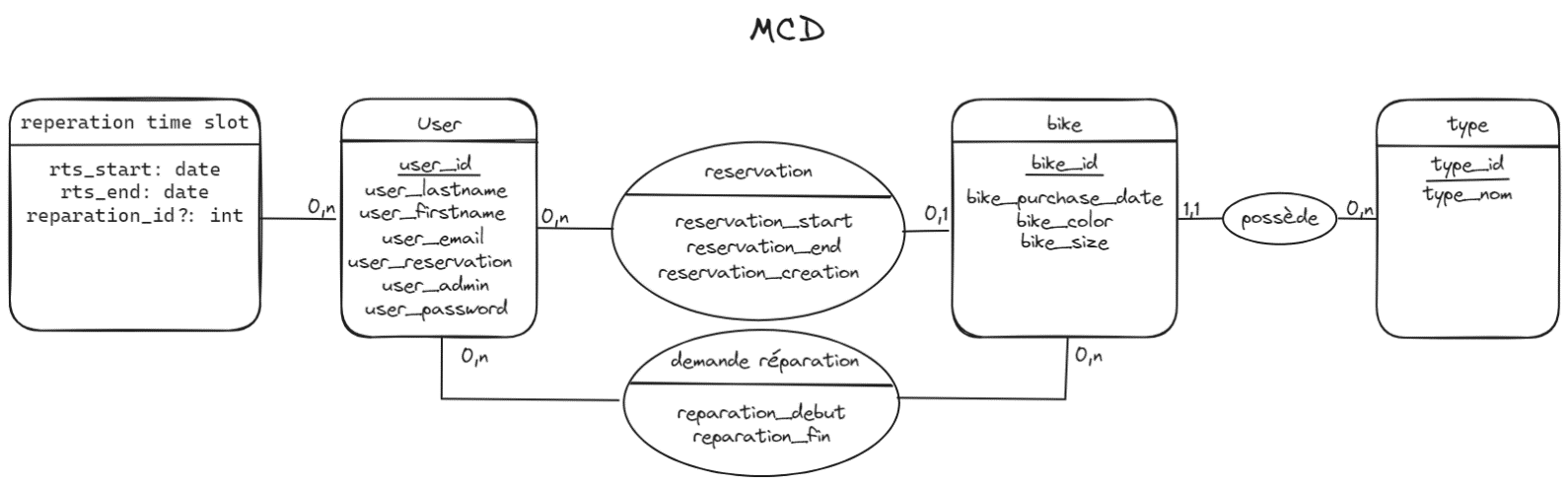


Figure 4 : MCD

## Base de données

Nous avons choisi d’utiliser phpMyAdmin pour créer la base de données et l’avons exporté en « .sql » Concernant la gestion des tables sont stockés en « InnoDB » et nous prenons le format « utf8mb4\_0900\_ai\_ci » qui est une appellation pour stipuler qu’on est en UTf-8 et aussi que la donnée est insensible à la casse et aux accents.

Une image contenant texte, Police, nombre, capture d’écran

Description générée automatiquement

Notre base de données s’intitule ismisparkg2. Dans celle-ci on aura 6 tables :

* Bike : elle reprend les différentes données de nos vélos comme « bike\_id » qui est la clé primaire de la table. Ensuite « bike\_type » notre foreign\_key qui sera reliée à la colonne type\_id dans la table « Type ». Aussi certaines autres données plus descriptives du vélo.
* Type : deux données l’une « type\_id » qui est la clé primaire de notre table et qui est reliée à la foreign\_key de la table « Bike », il y a aussi « type\_name » qui sera le type de vélo comme « VTT »
* User : on aura « user\_id » qui est la clé primaire ainsi que d’autre information concernant le nom et le prénom du client, son e-mail, son mot de passe hashé à l’aide de Bcrypt avec un coût de 10. Il y a également deux donnés en booléen, une qui est pour la réservation afin de voir si le client aura une réservation ou non, aussi une qui est pour vérifier si l’utilisateur est admin ou non afin de lui attribuer des fonctionnalités en plus.
* Reservation : la clé primaire sera « reservation\_id », aussi il y aura deux foreign\_key une « bike\_id » qui elle sera reliée à la table « Bike » et une autre « user\_id » reliée à la table « User ». Il y a aussi une donnée « reservation\_start » et « reservation\_end » pour stipuler les jours de début et de fin de réservation. Une donnée « reservation\_creation » sera nécessaire pour que l’admin voit quand la demande du client a été faite.
* Repair : la clé primaire sera « repair\_id » et trois foreign\_key seront présentes. Deux qui sont les mêmes que dans la table « Reservation » et pour les mêmes fonctionnalités c’est les données « bike\_id » et « user\_id ». La troisième sera « repair\_remplacement » elle est reliée à « bike\_id » dans la table « Bike », elle sera utilisée lorsque l’on devra donner un vélo de remplacement lors d’une réparation ainsi on verra quel vélo sera prêté via son id.
* Repair\_timeslot : sa clé primaire sera « rts\_id », deux foreign\_key seront nécessaire l’une « rts\_admin » qui sera reliée à « user\_id » dans la table « User » afin de vérifier si l’utilisateur sera admin ou non et ainsi lui donner certaines fonctionnalités. L’autre sera « reparation\_id » qui sera reliée à « repair\_id » dans la table « Repair ».

# Présentation de la partie Linux Server

## Partitionnement

* /home
  + Taille : 2Gio
  + Type : LVM
  + Système de fichier : ext4
* /
  + Taille : 7Gio
  + Type : LVM
  + Système de fichier : ext4
* /boot
  + Taille : 1024 Mio
  + Type : partition standard
  + Système de fichier : ext4
* /swap
  + Taille : 2Gio
  + Type : partition standard
  + Système de fichier : swap
* /web
  + Taille : 3Gio
  + Type : LVM
  + Système de fichier : ext4
* /backup (disque virtuel séparé
  + Taille : 5Gio
  + Type : LVM
  + Système de fichier : ext4

## Les rôles :

* Le serveur Web
* Héberge les pages Php
* Les images des différentes pages
* La base de données mysql
* La génération de clé SSL

## Packages Installés

* Httpd
* Php
* Php-Ldap
* Php-mysql
* Mod\_ssl
* Mysql
* Mysql-server mysql
* Iptables

## Sécurité

Niveau sécurité du serveur, nous avons verrouillé root afin qu’il soit impossible de s’y connecter directement. On a ajouté un compte admin en tant que sudouser. Pour les mots de passe nous en avons mis avec une complexité forte de 25 caractères contenant majuscules, minuscules et chiffres (sans caractères spéciaux par facilité pour le développement). Afin d’assurer une connexion sécurisée au site, nous avons utilisé le protocole HTTPS grâce au module SSL qui génère des certificatsQC par la suite signés par l’AD CS. Enfin on a changé le port pour la connexion SSH de 22 à 6969 et également mis en place un échangé de clé publiques.

## Backups

* Quoi
  + Les fichiers de configuration
  + La DB
  + La partition /web
* Où
  + Dans le /Backup
  + Prévu d’également sauver sur un NAS mais manque d’infrastructure
* Quand
  + Les fichiers de config une seule fois après modification
  + La DB et /web tous les jours à minuit
* Comment
  + Grace au crontab
* Vérification
  + Les sauvegardes sont vérifiées tous les jours mais pas tester par manque de temps

# Présentation de la partie Windows Server

Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquementPassons désormais à la partie de la gestion de Windows Server. Afin de gérer au mieux ce dernier, nous avons procéder étape par étape.  
 Dans un premier temps, nous avons créé *un Active Directory*, qui permettra de créer et de gérer facilement les utilisateurs/modérateurs de notre service.   
 Grâce à l’installation de l’Active Directory avant tout, *un serveur DNS* sera automatiquement créé, il n’y aura plus qu’a créer une Reverse Lookup Zone. Ce serveur DNS aura pour IP l’adresse de notre serveur en

Figure 5 : AD + DNS

l’occurrence : 192.168.100.1.   
Notre serveur DNS nous sera utile afin de simplifier l’accès à nos différents services.   
*Un alias (CNAME)* aura été créé afin de l’utiliser pour notre serveur web pour s’y connecter via l’adresse www.isims.park. Pour créer cet alias, il aura fallu créer un nouvel hôte (qui sera utilisé par le serveur web) qui a pour IP : 192.168.100.245 afin de pointer l’alias sur ce dernier.

Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquementPar la suite, nous avons créé notre *serveur DHCP* afin d’attribuer une adresse dynamiquement sur nos appareils à l’intérieur de notre réseau LAN. Notre serveur DHCP disposera d’un pool d’adresses allant de 192.168.100.2 à 192.168.100.254 avec un pool d’exclusions de 192.168.100.244 à 192.168.100.254 afin d’assurer de laisser une dizaine d’adresses disponibles pour quelconques routeurs/ machines, notre serveur web également.

Figure 6 : DHCP

Et enfin, nous avons procédé à l’implémentation d’un *service de backup* à l’aide du Tools Windows Server Backup. Pour ce faire, nous avons paramétré un backup journalier à 23 heures afin de s’assurer que personne ne soit connecté à cette heure. Nous avons au préalable créé un second disque dur (virtuel) de 20Go qui servira uniquement de disque de backup. Nous avons pris la décision de ne pas encombrer le disque de backup donc nous avons seulement paramétré le backup des fichiers les plus importants c-à-d : 1. System State, les fichiers NTDS & SYSVOL dans le dossier Windows système.

Une image contenant texte, capture d’écran, Police

Description générée automatiquementNous avons aussi pris la peine d’activer la poubelle de l’Active Directory. Cet élément est essentiel dans le cas où l’on supprime quelque chose, nous pouvons récupérer directement les éléments perdus.

Figure 7 : BackUp

# Problèmes rencontrés

Comme vous le savez très bien, en informatique, rien ne marche bien souvent du premier coup. C’est pour cela que nous avons recensé tous nos problèmes rencontrés lors de notre semaine de projet.

1. Problèmes lors de l'installation de l'AD notamment lors de la configuration des carte réseaux interne/externe. Ma carte réseau étant mal configurée en externe, le serveur ne communiquait pas avec le réseau LAN. Nous avons donc dû réinitialiser ma carte réseau virtuel et cela fonctionnait mieux.
2. Une grosse erreur que nous avons faite étant l’oubli de point de contrôle durant le projet, en effet, il est arrivé que notre serveur Windows ait crash suite à une mauvaise manipulation dans le service de backup et suite à cela, ce dernier a redémarré et s’est réinitialisé, on ne sait pas pourquoi...
3. Attention à bien être sûr et certain à désactiver les protocoles IPV6 des différents serveurs et machines sinon cela posait des problèmes notamment lors des PING, NSLOOKUP.
4. Il était important de laisser actif uniquement les cartes réseaux nécessaires au bon fonctionnement du réseau LAN sinon la connectivité pouvait prendre le dessus selon les cartes réseaux activés.
5. Désactiver le DHCP sur l’interface réseau de linux en IPV4.
6. Changer le Virtual network Editer sur VMWARE, le passer le bridge sur le port Ethernet au lieu de le mettre en automatique !
7. Afin de simplifier l'exportation d'un fichier de notre Certification d’authentification depuis notre serveur Windows Server vers notre serveur linux, nous avons donc dû créer un dossier de partage afin de récupérer le fichier sur le réseau pour ensuite l'importer vers notre serveur web linux.
8. Quand le SSH ne fonctionne pas, il suffit de réinitialiser les paramètres, enlever le DHCP et ensuite tout remettre correctement et c’est censé fonctionner.
9. Problème des options du LDAP qui nous a demandé un débogage.
10. Problèmes de configuration du Firewall sur le serveur Linux, les paquets pour le serveur LDAP qui étaient bloqués.

# Conclusion

Durant ce projet interdisciplinaire, nous avons, par rapport à l’année passée, appris beaucoup plus de choses, autant sur le plan pratique que sur le plan relationnel.

En général, nous ne nous connaissions pas, il a donc été nécessaire dès le début de s’entendre afin de pouvoir travailler dans la bonne entente. Nous avons su dès le début se partager les tâches et s’organiser de manière équitable. Malgré les faiblesses de chacun, nous avons su les combler avec les forces des autres.

Concernant l’apprentissage lors de ce projet, beaucoup de lacunes ont été comblées pour tout le monde, par le savoir des uns mais aussi par le biais de recherches sur diverses plateformes pour pouvoir avancer. C’est là que l’on a appris à différencier aussi les bonnes et mauvaises informations, car avoir un aspect critique sur les données est aussi important.

En quelques mots, cette expérience a été fabuleuse, certes pleine de stress et un manque de temps pour terminer certaines fonctionnalités mais très enrichissante.

# Bibliographie

Sites utilisés et logiciels :

- cours : https://ecampus.heh.be/course/view.php?id=2494

- énoncés du projet : https://hehbe.sharepoint.com/:w:/r/sites/ProjetInterdisciplinaireEtudiants-Gr3/\_layouts/15/Doc2.aspx?action=edit&sourcedoc=%7B7f98c840-cfe4-4c5d-9439-cb088477fad5%7D&wdOrigin=TEAMS-MAGLEV.teamsSdk\_ns.rwc&wdExp=TEAMS-TREATMENT&wdhostclicktime=1702885744090&web=1 et https://hehbe.sharepoint.com/:w:/r/sites/ProjetInterdisciplinaireEtudiants/\_layouts/15/Doc2.aspx?action=edit&sourcedoc=%7B43148f2c-4b82-40a0-adda-bc8e45c72766%7D&wdOrigin=TEAMS-MAGLEV.teamsSdk\_ns.rwc&wdExp=TEAMS-TREATMENT&wdhostclicktime=1703231996190&web=1

- chatgpt : https://chat.openai.com

- https://css-tricks.com/blobs/

- [https://cssgradient.io/gradient-backgrounds/](https://cssgradient.io/gradient-backgrounds/ "https://cssgradient.io/gradient-backgrounds/")

- [https://www.svgrepo.com/svg/70289/bike](https://www.svgrepo.com/svg/70289/bike "https://www.svgrepo.com/svg/70289/bike")

- [https://www.figma.com/file/JblVXJ26Wte8KuQqxJ7Wkm/ISIMSpark?type=design&node-id=0-1&mode=design&t=KxJfaqgrgNXwBirM-0](https://www.figma.com/file/JblVXJ26Wte8KuQqxJ7Wkm/ISIMSpark?type=design&node-id=0-1&mode=design&t=KxJfaqgrgNXwBirM-0 "https://www.figma.com/file/JblVXJ26Wte8KuQqxJ7Wkm/ISIMSpark?type=design&node-id=0-1&mode=design&t=KxJfaqgrgNXwBirM-0")

- [https://fonts.google.com/icons?selected=Material+Symbols+Outlined:pedal\_bike:FILL@0;wght@400;GRAD@0;opsz@24&icon.query=bike](https://fonts.google.com/icons?selected=Material+Symbols+Outlined:pedal\_bike:FILL@0;wght@400;GRAD@0;opsz@24&icon.query=bike "https://fonts.google.com/icons?selected=Material+Symbols+Outlined:pedal\_bike:FILL@0;wght@400;GRAD@0;opsz@24&icon.query=bike")

- https://almalinux.org

- https://code.visualstudio.com

- https://www.phpmyadmin.net

- https://discord.com

- https://code.visualstudio.com

- https://github.com/ADL1337/Projet-Interdisciplinaire-G2/blob/main/rapport/Cahier\_de\_bord\_linux.md

- https://www.microsoft.com/fr-be/microsoft-365/word?market=be

- https://laragon.org/index.html et https://www.wampserver.com

- https://www.microsoft.com/fr-be/microsoft-teams/log-in

Les tuto utilisés :

- [https://gridpane.com/kb/generate-an-ssh-key-with-termius/](https://gridpane.com/kb/generate-an-ssh-key-with-termius/ "https://gridpane.com/kb/generate-an-ssh-key-with-termius/")

- https://ubunlog.com/fr/s-search-busca-web-desde-terminal/

- https://access.redhat.com/documentation/fr-fr/red\_hat\_enterprise\_linux/9/html/configuring\_firewalls\_and\_packet\_filters/using-and-configuring-firewalld\_firewall-packet-filters#firewalld-zones\_getting-started-with-firewalld

- https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-using-firewalld-on-centos-8-fr

- https://unix.stackexchange.com/questions/367017/updating-network-manager

- https://access.redhat.com/documentation/fr-fr/red\_hat\_enterprise\_linux/9/html/configuring\_firewalls\_and\_packet\_filters/viewing-the-current-status-and-settings-of-firewalld\_using-and-configuring-firewalld#:~:text=L%27option%20--list-all,et%20à%20la%20connexion%20actives.

- https://grafikart.fr/tutoriels/iptables-694

- https://www.redhat.com/sysadmin/iptables

- https://www.vinchin.com/en/blog/backup-and-restore-linux-server.html

- https://www.howtoforge.com/bash-script-for-configuring-iptables-firewall

- https://support.termius.com/hc/en-us/articles/4401872025113-Keychain

- https://gridpane.com/kb/generate-an-ssh-key-with-termius/

- https://www.digitalocean.com/community/tutorial-collections/how-to-protect-ssh-with-fail2ban

- https://serverfault.com/questions/135906/when-does-cron-daily-run

- https://openclassrooms.com/forum/sujet/lier-deux-pages-php-avec-un-bouton

- https://doc.ubuntu-fr.org/creer\_un\_service\_avec\_systemd

Cours de l'école

- Dev backend PHP : https://ecampus.heh.be/course/view.php?id=2369

- Linux : https://ecampus.heh.be/course/view.php?id=2740

- Windows server : https://ecampus.heh.be/course/view.php?id=2834