



RAPPORT D'ACTIVITE PROFESSIONNELLE

Licence Informatique option cybersécurité

Luc-Louis DELORME

Sommaire

1 – Présentation de l’entreprise	2
1.1 – Historique.....	2
1.2 – Activité	2
1.3 – Organigramme	3
1.4 – Service informatique	4
1.5 – Sécurité des données et RGPD	4
2 – Présentation d’une activité journalière	5
2.1 – Récupération des données.....	5
2.2 – Importation des données.....	8
2.3 – Commandes SQL	9
2.4 – Script Python	10
3 – Présentation des activités secondaires	12
3.1 – Inventaire du matériel.....	12
3.2 – Assistance technique.....	12
3.3 – Mise à niveau des postes de travail	12
4 – Projet d’entreprise	13
5 – Remerciements	14
6 – Annexes	15

1.4 – Service informatique

La société est équipée d'une salle informatique de 30 postes fonctionnant sous Windows 10 professionnel. Ces postes sont maintenus et remplacés par la société MC2I. Lorsqu'une machine est hors-service, le disque dur est récupéré et détruit physiquement par un broyeur de disques durs. Des postes téléphoniques analogiques Alcatel Temporis 700, des micro-casques AxTel Pro Duo Wideband IP avec câbles RJ9 sont également mis à disposition des agents commerciaux. Les serveurs sont au nombre de trois : un serveur d'appel « ACD Discofonée » fonctionnant sur un PABX AASTRA 5000, Un serveur de fichier Active Directory Windows pour la mise en commun des ressources de travail et la gestion des droits d'accès, et un serveur d'écoute de conversation téléphonique ALL RTP. Le pare-feu matériel ZyXel USG60W configuré par le constructeur et maintenu par la société MC2I sert à bloquer l'intégralité des paquets entre le réseau interne et externe, et sécurise le réseau local par le flux entrant. Seuls les sites Internet et les ports nécessaires à l'accomplissement des missions sont autorisés. Le pare feu bloque les malwares et les intrusions grâce à un système de détection. Il est également utilisé comme VPN TLS.

1.5 – Sécurité des données et RGPD

Dans le cadre de la loi du Règlement général sur la protection des données (RGPD), la CNIL s'est vue conférée depuis le 25 mai 2018 le pouvoir de sanctionner un organisme qui ne respecte pas ce règlement. Celui-ci a pour but de protéger les données personnelles de chaque individu, en particulier dans l'espace digital. Tout individu a le contrôle sur ses données et a le droit de demander l'accès, la modification ou la suppression de ses données. Respecter ce règlement va dans la continuité des bonnes pratiques informatiques en termes de sécurité. Pour FlexyCall, les données personnelles d'une personne ne peuvent être détenues que suivant son accord, ou l'accord de la société à qui le client a communiqué ses données volontairement.

Le service informatique de FlexyCall est très rigoureux dans le respect des règles de sécurités préconisées par le RGPD. De nombreuses mesures de sécurité sont appliquées, comme la mise à jour des logiciels ou des systèmes d'exploitation, l'utilisation d'antivirus sur chaque poste client, le paramétrage de connexion au wifi en wpa2-psk ou wpa3-sae, le paramétrage d'identifiants et de mots de passes forts pour chaque compte de connexion.

Dans la conception du code PHP pour l'intranet, d'autres mesures sont appliquées, comme la préparation de requête SQL, l'utilisation de bibliothèques cryptographiques reconnues (comme PHPMailer pour l'envoi d'e-mails), l'utilisation des requêtes POST pour le transfert des données dans l'intranet et la veille technologique.

2 – Présentation d’une activité journalière

2.1 – Récupération des données

Le client dépose sur le serveur SFTP de l’entreprise un fichier CSV contenant une liste de contacts à démarcher. Celle-ci contient de nombreuses données confidentielles (noms, prénoms, adresses...) Ces données doivent être structurées pour être insérées dans les bases de données locales de l’entreprise.

Le logiciel Filezilla me permet d’accéder au serveur avec le protocole SFTP. Je récupère le fichier du client dans un répertoire que je titre à la date du jour. J’ajoute dans ce répertoire 4 fichiers modèles pour stocker les données. Ces fichiers sont ensuite envoyés dans plusieurs tables de données différentes :

- La table de l’ACD « Discophonee » pour le CRM des agents. Les informations sur le client contacté sont visibles à l’écran. Les agents peuvent ajouter des qualifications ou des commentaires à l’aide d’une interface graphique écrite en PHP.
- La table de l’automate d’appel. Celui-ci programme des appels vers les contacts enregistrés de façon journalière, jusqu’à 5 fois maximum par contact.
- La table de suivi de mission dans la base de données de l’entreprise. Cela permet aux techniciens de suivre et de contrôler les contacts,
- Le fichier « contacts Call Center » est transmis directement aux agents. Il contient des numéros spéciaux qui ne sont pas pris en charge par l’automate d’appel, ils sont donc contactés manuellement avec un softphone connecté à la base de données.

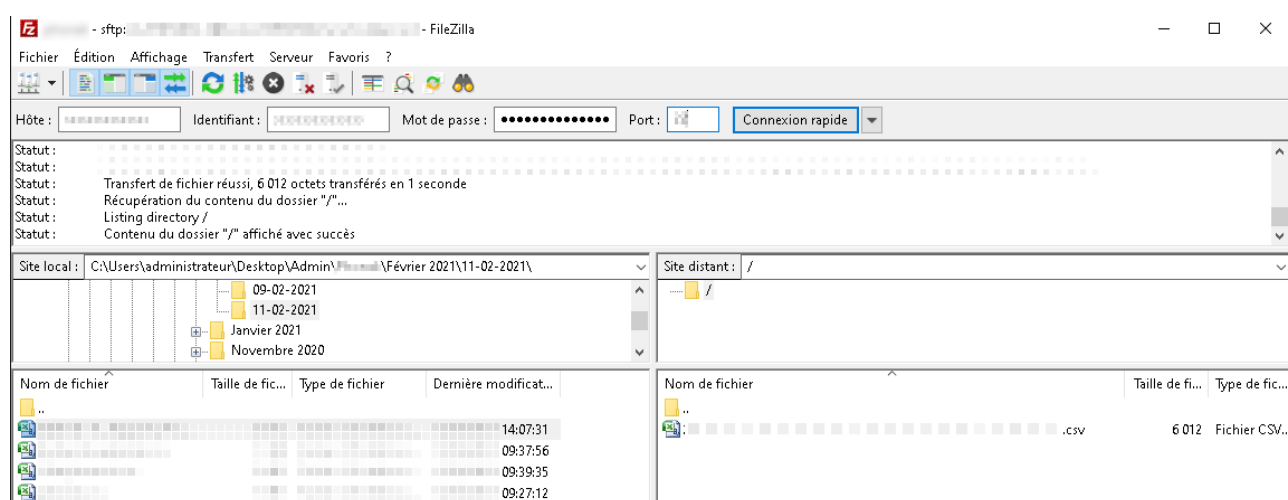


Figure 2

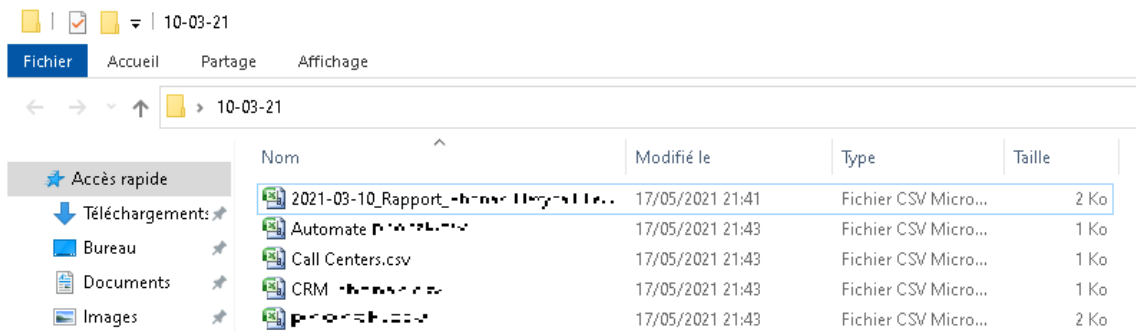


Figure 3

Dans le fichier **source** (intitulé : "2021-03-10_Rapport_mission..."), je rajoute deux colonnes : l'une contient un numéro de ligne et l'autre l'identifiant unique. Pour l'obtenir, je concatène l'année, le jour, le mois, le numéro de ligne et le numéro de la mission : {AAAA-MM-JJ}_{Ligne}_{Mission}. Je renseigne le premier fichier type « **mission.csv** ». Celui-ci sera injecté dans la table de suivi de mission de l'entreprise. Je copie les données du fichier source dans ce fichier en veillant à respecter la correspondance entre les colonnes :

	A	B	C	D	E	F	G	H	I	J
1	n	id	Surname	Name	Email address	Phone num	Statut de la	Propriétaire	MC Form Id	Lead Cr
2		1 210310_1_26	Joël	Harmon	joel.harmon@...	068700441	Open			11/02/
3		2 210310_2_26	Daniel	Ber...	daniel.ber...	061770181	Open			11/02/
4		3 210310_3_26	Alain	Ber...	alain.ber...	063476120	Open			11/02/
5		4 210310_4_26	Jean-Yves	Gaut...	jean.yves.gau...	073683898	Open			11/02/
6		5 210310_5_26	Dominique	...	dominique....	064636780	Open			11/02/
7		6 210310_6_26	Anna	Jar...	anna.jar...	061030871	Open			11/02/
8		7 210310_7_26	Tess	Pat	tess.pat...	074825820	Open			11/02/
9		8 210310_8_26	Frédéric	Fel...	frederic.fel...	071461084	Open			11/02/
10		9 210310_9_26	Gerard	Fou...	gerard.fou...	068904038	Open			11/02/

Figure 4

Dans le fichier **source**, je rajoute une colonne à côté des numéros de téléphone. La mise en forme des numéros reçus (11 chiffres avec le préfixe) doit correspondre à 9 chiffres (sans préfixe). Pour cela, j'utilise la fonction « =droite » d'Excel. J'ouvre le second fichier modèle. Je copie les données du fichier **source** dans le deuxième fichier « **CRM.csv** » en veillant à respecter la correspondance entre les colonnes. Ces données sont exploitées par le GRC (CRM) des agents, qui affiche à l'écran les informations sur le client contacté.

M	N	O	P	Q	R	S	T
HCP Postal C	HCP Phone N	numéros	HCP address	HCP city	HCP email address		
	3,3252E+10	=DROITE(N2;9)*1					
	3,3473E+10						
	3,3982E+10						
	3,3564E+10						
	3,324E+10						
	3,3324E+10						
	3,3494E+10						
	3,3139E+10						
	3,3243E+10						
	3,3243E+10						
	3,349E+10						
	3,3429E+10						
	3,3327E+10						
	3,3564E+10						
	3,3473E+10						
	3,3322E+10						

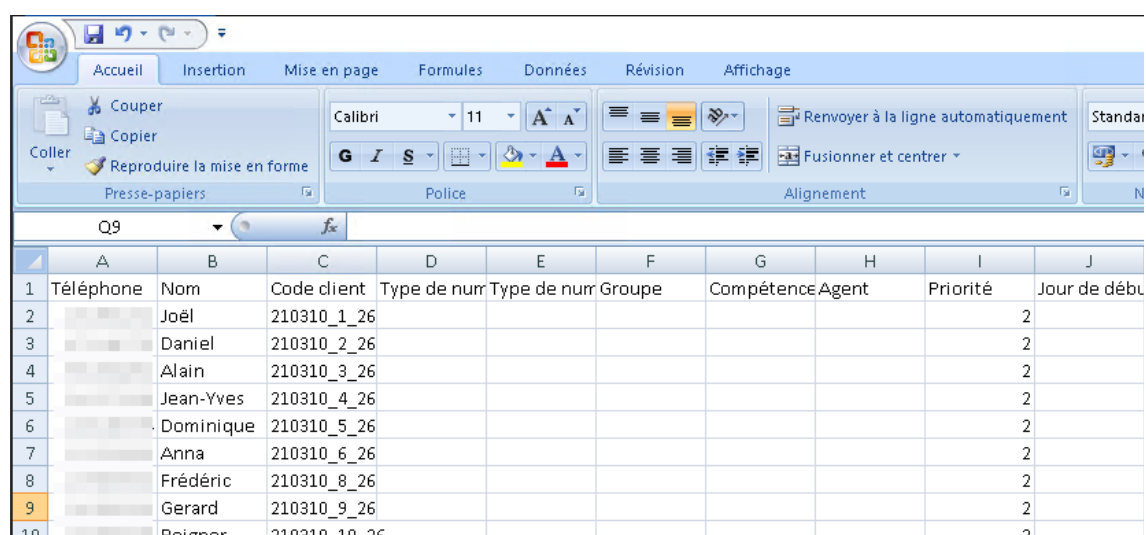
Figure 5

Dans un troisième temps, j'isole les numéros spéciaux à l'aide d'un tri automatique. Je mets l'indice de ces numéros en rouge pour les faire remonter lors d'un tri par couleur. Je les copie ensuite dans le troisième fichier « **Contacts Call center.csv** » :

R14							
	A	B	C	D	E	F	G
1	n	ID	Surname	Name	Email address	Phone num	Statut de la p
2	7	210310_7_26	Tess				
3	16	210310_16_26	Catherine				
4	21	210310_21_26	Jacques				
5	22	210310_22_26	Suzanne				
6	23	210310_23_26	Thierry				
7	24	210310_24_26	Jean-Luc				
8	28	210310_28_26	Dubois				
9	1	210310_1_26	Joël				

Figure 6

Enfin, je copie les identifiants, les noms et les numéros restants dans le dernier fichier « **automate.csv** ». Je renseigne 2 comme priorité. La valeur la plus faible étant prioritaire, toutes les personnes seront contactées dans les 3 jours :

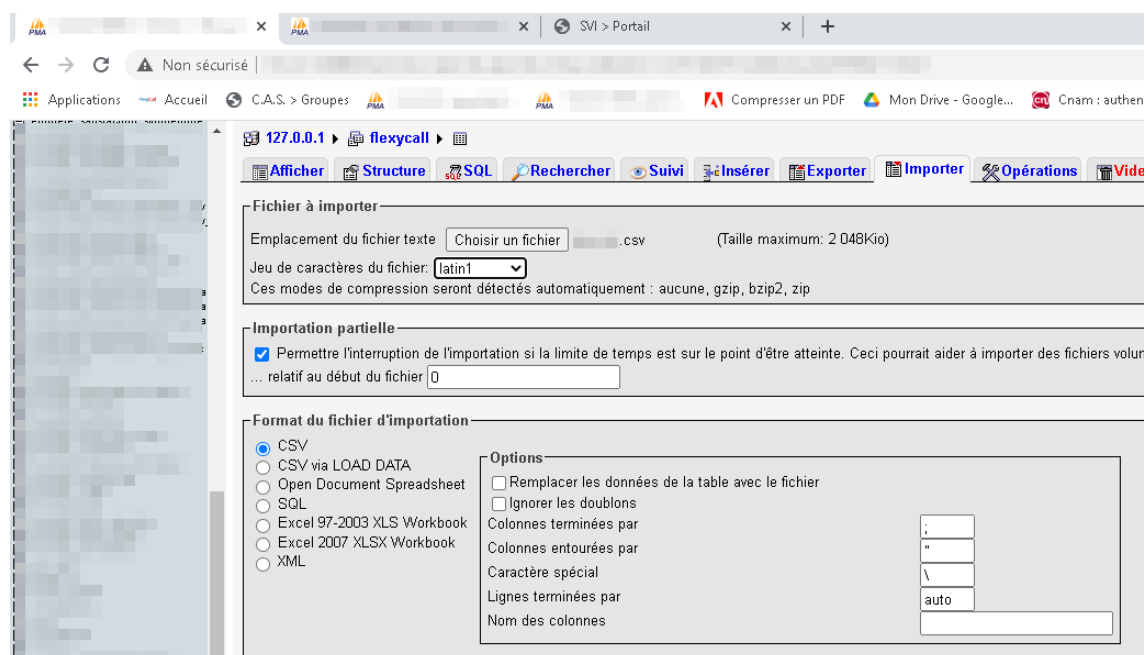


	A	B	C	D	E	F	G	H	I	J
1	Téléphone	Nom	Code client	Type de num	Type de num	Groupe	Compétence	Agent	Priorité	Jour de début
2		Joël	210310_1_26						2	
3		Daniel	210310_2_26						2	
4		Alain	210310_3_26						2	
5		Jean-Yves	210310_4_26						2	
6		Dominique	210310_5_26						2	
7		Anna	210310_6_26						2	
8		Frédéric	210310_8_26						2	
9		Gerard	210310_9_26						2	
10		Reimer	210310_10_26						2	

Figure 7

2.2 – Importation des données

Les fichiers sont importés dans 3 tables différentes. Le fichier **mission.csv** est injecté dans la table de suivi de mission dans base de données du serveur principal. Je sélectionne le fichier et le jeu de caractère puis je clique sur « Exécuter » :



Fichier à importer

Emplacement du fichier texte : Choisir un fichier .csv (Taille maximum: 2 048Kio)

Jeu de caractères du fichier: latin1

Ces modes de compression seront détectés automatiquement : aucune, gzip, bzip2, zip

Importation partielle

☒ Permettre l'interruption de l'importation si la limite de temps est sur le point d'être atteinte. Ceci pourrait aider à importer des fichiers volumineux...

Format du fichier d'importation

☒ CSV

☐ CSV via LOAD DATA

☐ Open Document Spreadsheet

☐ SQL

☐ Excel 97-2003 XLS Workbook

☐ Excel 2007 XLSX Workbook

☐ XML

Options

☐ Remplacer les données de la table avec le fichier

☐ Ignorer les doublons

Colonnes terminées par : ;

Colonnes entourées par : "

Caractère spécial : \

Lignes terminées par : auto

Nom des colonnes :

Figure 8

Le fichier **CRM.csv** est importé dans une table du serveur d'appel ACD.

Le fichier **automate.csv** est importé dans une autre table de l'ACD. Je choisis le fichier, ainsi que le groupe d'importation, et je valide en cliquant sur « Exécuter » :

The screenshot shows the 'Importation MANUELLE' (Manual Import) section of the discofonee CRM interface. The interface has a top navigation bar with links: Portail, SVI, Messagerie vocale, A.C.D., C.A.S., Maintenance, Avancé, and Administration. A left sidebar contains a tree view with categories like Général, Qualifications, Jours exceptionnels, Fonctions avancées, Groupes, Contacts, Import/export contacts (highlighted), Types de numéros, and Journal. The main content area is titled 'Importation MANUELLE' and includes instructions: 'Pour insérer votre fichier, créez-le sous excel, puis, enregistrez-le au format CSV : CSV (séparateur:point-virgule) (*.csv)'. It contains three numbered steps: 1. 'Indiquer le groupe d'importation:' with a dropdown menu set to '26'; 2. 'Indiquez le type d'insertion:' with radio buttons for 'en complétant les données existantes' (selected) and 'en supprimant les données existantes'; 3. 'Indiquez l'adresse de votre fichier' with a 'Choisir un fichier' button and a text field containing 'Automate.csv'. Below these steps are 'Importer' and 'Retour' buttons. A second section titled 'Exportation' has a checkbox 'Exporter le nom des champs sur la première ligne' (unchecked) and a radio button 'Exporter la table entière' (selected), with 'Exporter' and 'Retour' buttons at the bottom.

Figure 9

2.3 – Commandes SQL

Dans le cadre de mon apprentissage, j'ai dû concevoir des requêtes SQL pour trier et rechercher des informations dans la base de données en fonction de critères précis. Cette requête m'a permis de trouver les dossiers clients non clôturés :

```
SELECT * FROM `table`  
WHERE (  
  `qualification` = "Refus de RDV: Client n'a pas eu le temps d'appeler"  
  OR `qualification` = "Refus de RDV: Lead injoignable"  
  OR `qualification` = "Refus de RDV: Lead non reçu"  
  OR `qualification` = "Client pas disponible (déjà en rdv)"  
  OR `qualification` = "Client fermé, hors horaires/jours d'ouverture"  
  OR `qualification` = "Répondeur"  
  OR `qualification` = "Barrage assistante"  
)  
AND `tentatives` < '5'  
group by `id_contact`, `surname`, `name`
```


Cette requête permet de trier les dossiers clôturés pour les enlever du CRM des agents. Ceci leur permet d'éviter de rappeler plusieurs fois les mêmes clients lorsque les dossiers sont clôturés :

```
SELECT `id_contact`, `name`, `surname`  
FROM `table`  
WHERE (  
  `qualification` = 'Refus de RDV: Lead pas intéressé: préfère voir son médecin ORL avant'  
  OR `qualification` = 'Refus de RDV: Lead pas intéressé: simple curiosité'  
  OR `qualification` = 'Refus de RDV: Lead faux numéro'  
  OR `qualification` = 'Rdv pris'  
  OR `qualification` = 'Refus de RDV: Lead pas intéressé: souhaitait juste de l'info mais pas de rdv'  
  OR `qualification` = 'Faux numéro'  
  OR `tentatives` = '5'  
)  
AND `id_contact` LIKE ('2102%')
```

La requête suivante permet d'afficher le nombre d'agents connectés et prêt à recevoir des appels pour chaque mission. La lettre « S » signifie « *En service* » :

```
SELECT COUNT(distinct agent) AS agent_dispo FROM table where bdate = curdate() and  
heure >= '09:00:00' and heure <= '19:00:00' and (groupe like ('14,%') or groupe like  
('%14,%') or groupe like ('%,14')) and action_ = 'S'
```

2.4 – Script Python

Ce script permet d'envoyer une alerte par mail lorsqu'un nouveau fichier à traiter est disponible sur le serveur sftp. Il fonctionne avec la version 3.8 et avec les librairies pysftp, smtplib, ssl et MIMEText.

Pysftp est un module simple d'utilisation permettant d'effectuer des tâches sur un serveur disposant d'un mode de connexion SFTP. Le module smtplib permet d'envoyer des emails en définissant un objet client SMTP. Le module SSL permet de chiffrer les flux de données entre le client et le serveur à l'aide des protocoles SSL ou TLS. Enfin, le module MIMEText permet de créer et de personnaliser un objet message puis de l'inclure dans le corps du mail :

```
#!/usr/bin/python3  
  
import pysftp, smtplib, ssl  
from email.mime.text import MIMEText  
  
with pysftp.Connection('xxx.xxx.xxx.xxx', username='identifiant', password='password') as  
sftp:
```

```

files = len(sftp.listdir())

if files > 0:
    smtp_server = 'smtp.ionos.fr'
    port = 587 # Chiffrement TLS
    destinateur = 'robot@flexycall.fr'
    destinataire = 'ldelorme@flexycall.fr'
    password = 'motdepasse'
    msg = "Il y a " + str(files) + " fichier(s) en attente."

    message = MIMEText(msg, _charset='UTF-8')
    message['Subject'] = 'Rapport de mission !'
    message['From'] = destinateur
    message['To'] = destinataire

    context = ssl.create_default_context()
    with smtplib.SMTP(smtp_server, port) as server:
        server.starttls(context=context)
        server.login(destinateur, password)
        server.sendmail(destinateur, destinataire, message.as_string())
        server.quit()

sftp.close()

```

Une connexion s'établit d'abord au serveur FTP avec le protocole SSH, et le nombre de fichier disponible est enregistré dans une variable. Si le nombre des fichiers est supérieur à 0, alors la condition s'active : une connexion au compte gmail s'établit avec un chiffrement TLS et un mail est envoyé au destinataire. Une fois le processus terminé, toutes les sessions sont déconnectées. J'automatise la tâche à l'aide du daemon cron sur un système Linux :

```
0 9,10,11,14,15,16 * * 1-4 /home/user/alertemail.py
```

Le programme s'exécute donc à 9, 10, 11, 14, 15 et 16 heures du lundi au jeudi.

3 – Présentation des activités secondaires

3.1 – Inventaire du matériel

J'ai répertorié et tenu à jour l'inventaire du matériel informatique disponible. Cela m'a permis de mieux connaître les équipements utilisés ainsi que leur spécificité, mais également d'être plus réactif en cas de problème matériel. Parmi ce matériel, il y a des souris, des claviers, des écrans avec connecteurs VGA, DisplayPort ou HDMI, des casques Axtel avec embout USB, des unités centrales, des clés 4G, des adaptateurs DisplayPort vers VGA ou HDMI vers VGA, des webcams et des câbles RJ45 catégorie 6a ou 5a s/ftp de différentes longueurs.

3.2 – Assistance technique

J'ai participé à la résolution de problèmes techniques rencontrés par les agents, comme des problèmes d'accès à internet ou au réseau d'entreprise, des dysfonctionnements de logiciels métiers, des dysfonctionnements des périphériques matériels (souris, claviers, casques), ou des demandes d'accès à des ressources partagées. Les problèmes techniques doivent être résolus le plus rapidement possible pour éviter une perte d'activité dommageable pour les agents ou l'entreprise. Cela exige donc une bonne connaissance du système Windows et une bonne utilisation des moteurs de recherche pour trouver l'information rapidement.

3.3 – Mise à niveau des postes de travail

J'ai également opéré la mise à niveau des PC fonctionnant avec le système d'exploitation Windows 7 vers Windows 10. Pour cela, j'ai utilisé Windows Media Creation Tool. J'ai veillé au bon fonctionnement des composants de l'ordinateur (comme les disques durs) à l'aide du logiciel Hwinfo, avant d'effectuer la migration. Enfin, j'ai contrôlé la bonne intégration des logiciels et le bon fonctionnement des accès réseau sur le nouveau système, et réinstaller manuellement les logiciels en cas de nécessité.

4 – Projet d'entreprise

Mon maître de stage souhaite moderniser la plateforme de développement web de l'intranet, et reprogrammer le code PHP de certaines pages pour le rendre conforme aux normes RGPD. Chaque agent doit avoir accès à un espace personnalisé via une interface de connexion sécurisée. La modernisation inclut la base de données MySQL, l'interface web d'administration PHPMyAdmin, le serveur web Apache et le langage de programmation PHP.

De nombreux environnements de développement ont été testés sur des machines virtuelles Windows 10 pour déterminer celui qui correspond le mieux aux besoins de l'entreprise. En effet, le logiciel sur lequel se baser pour construire le nouvel intranet doit être fiable, sécurisé, efficient, maintenu dans le temps, gratuit, et adapté à l'utilisation professionnelle. Le déroulement du projet s'effectue en plusieurs phases :

- La recherche du meilleur logiciel de développement. Pour cela, il est possible de s'appuyer sur les réseaux spécialisés, comme Github, Stack Overflow, Reddit ou des blogs maintenus par des développeurs.
- L'installation du logiciel sur le serveur. Des runtimes, ou environnements d'exécution, doivent parfois être installés manuellement. Il faut également veiller à ce que la configuration des nouveaux modules n'entre pas en conflit avec celle déjà existante (comme le port par défaut ou le dossier d'installation du serveur web et de la base de données.)
- La migration des tables de la base de données dans le nouveau système de gestion des bases de données. La migration doit être progressive pour faciliter l'intégration du logiciel dans l'environnement informatique de l'entreprise. De plus, la migration doit s'effectuer à chaud, alors que l'entreprise continue de fonctionner.

Forces : <ul style="list-style-type: none">• Améliorer la vitesse d'exécution des codes PHP en passant à la version 7 ou 8,• Obtenir de nouvelles fonctionnalités PHP (chiffrement TLS pour les mails)	Faiblesses : <ul style="list-style-type: none">• Migration progressive,• Migration « à chaud »,• Difficulté pour choisir un logiciel de développement.
Opportunités : <ul style="list-style-type: none">• Améliorer la productivité• Améliorer la sécurité	Menaces : <ul style="list-style-type: none">• Risque de devoir effectuer une autre migration si l'équipe de développement du logiciel choisi renonce au projet.

Après avoir hésité entre les logiciels WNMP (Nginx 1.21.0 – MariaDB 10.5.10 – PHP 8.0.6 FastCGI – phpMyAdmin 5.1.0) et WAMP (Apache 2.4.41 – PHP 5.6.40, 7.3.12, 7.4.0 – MariaDB 10.4.10|10.3.20 – PhpMyAdmin 4.9.2). Mon maître de stage s’est orienté plutôt vers WAMP pour sa popularité, son ancienneté et sa communauté d’utilisateurs.

<https://wnmp.x64architecture.com/>

<https://www.wampserver.com/>

5 – Remerciements

Je tiens à remercier M. FABRU, M. MALEYSSON et M. DUPRE, pour m’avoir accordé la chance d’apprendre le métier d’informaticien au sein de la société FlexyCall. Cette année fut pour moi riche en apprentissage. En effet, j’ai pu acquérir de nombreuses compétences professionnelles dans le domaine de la programmation, de la sécurité et de l’assistance technique.

6 – Annexes

Ce script en Python (sur Windows) me permet d'automatiser en grande partie ma tâche journalière :

```
import pysftp, os, time, csv

# Options :
targetPath = '/'

# Changer la date du mois dans cette variable (créer le dossier avant) :
downloadPath = 'C:/Users/administrateur/Desktop/Dossiers/Mission/Mai 2021/'

dossier = time.strftime("%d-%m-%Y")
destination = downloadPath + dossier
dateJour = time.strftime("%Y-%m-%d 12:00:00")
jeton = False

#~* | Etape 1 : Téléchargement du fichier | *~#

print(" ")
if os.path.isdir(downloadPath + dossier) == False:
    os.mkdir(downloadPath + dossier)

with pysftp.Connection('xxx.xxx.xxx.xxx', username='identifiant', password='password') as sftp:
    print("Connexion TLS établie ...")
    print(" ")
    files = len(sftp.listdir())
    if files == 1:
        jeton = True
    try:
        nomFichier = sftp.listdir()
        fichier = nomFichier[0]
        sftp.get_d(targetPath, destination, preserve_mtime=True)
        sftp.remove(fichier)

    except:
        print("Erreur try 1 ... ")
        print(" ")
        sftp.close()

#~* | Etape 2 : Création des fichiers de production | *~#

tableau = []

# Extraction des données du fichier source :
if jeton == True:
```

```

try :
    print("Création des fichiers ...")
    print(" ")
    csvfile = open(destination + '/' + fichier, encoding='latin-1', newline='')
    for row in csv.reader(csvfile, delimiter=','):
        tableau.append(row)
except :
    print("Erreur try 2")
    print(" ")

# Création du tableau pour le fichier mission.csv :
contacts = []
i = 0
limite = len(tableau)
for row in tableau:
    contacts.append(["0", "Mission Sortant J+3", dateJour, "", tableau[i][2], tableau[i][1],
        tableau[i][3], tableau[i][4], tableau[i][5], tableau[i][6], tableau[i][7], tableau[i][8],
        tableau[i][10], tableau[i][9], tableau[i][11], tableau[i][12], tableau[i][13],
        tableau[i][14], tableau[i][15], tableau[i][0]])
    i = i + 1

# Création du code client :
i = 0
timeCode = time.strftime("%y%m%d")
for row in contacts:
    row[3] = str(timeCode) + "_" + str(i) + "_26"
    i = i + 1

# Corriger les numéros de téléphones :
for row in contacts:
    row[15] = row[15].replace(" ", "")

# Correction de la colonne lead_creation_date:
leadCreationDate = time.strftime("%d/%m/%Y")
for row in contacts:
    if (row[11] == '44333') or (row[11] == '44334'):
        row[11] = leadCreationDate

# Suppression de la ligne 1 portant les noms de colonnes : (name, surname...)
del contacts[0]

# Création du fichier mission.csv :
try :
    mission = open(destination + '/mission.csv', 'w', encoding='latin-1', newline='')
    ecrire = csv.writer(mission, delimiter=',')
    for ligneContact in contacts:
        ecrire.writerow(ligneContact)
except :

```

```

    print("Erreur try 3")
    print(" ")

# Création du tableau pour le fichier CRM.csv :
contactsCRM = []
i = 0
for row in contacts:
    contactsCRM.append(["0",dateJour,contacts[i][4],contacts[i][5],contacts[i][12],
        contacts[i][14],contacts[i][17],contacts[i][3],"",contacts[i][15], "", "", "",
        contacts[i][19], "0000-00-00",contacts[i][1], "", "0", "0000-00-00"])
    i = i + 1

# Traitement des numéros pour le crm (récupération des numéros à partir de la droite:
for row in contactsCRM:
    row[9] = row[9][-9:]

# Création du fichier CRM.csv :
try:
    crm = open(destination + '/CRM phonak.csv', 'w', encoding='latin-1', newline='')
    ecrire = csv.writer(crm, delimiter=',')
    for ligneContact in contactsCRM:
        ecrire.writerow(ligneContact)

except :
    print("Erreur try 4")
    print(" ")

# Création du tableau pour automate.csv :
contactsAutomate = []

i = 0
for row in contactsCRM:
    if (row[9] != '56*****31') and (row[9] != '#N/A'):
        row[9] = "0" + row[9]
        contactsAutomate.append([contactsCRM[i][9],contactsCRM[i][2],
            contactsCRM[i][7], "", "", "", "", "", "2", "", "", ""])
        i = i + 1

# Création du fichier automate.csv :
try :
    automate = open(destination + '/Automate.csv', 'w', encoding='latin-1', newline='')
    ecrire = csv.writer(automate, delimiter=',')
    for ligneContact in contactsAutomate:
        ecrire.writerow(ligneContact)

except :
    print("Erreur try 5")
    print(" ")

```



```

# Création du fichier callcenters.csv :
colonnesCallCenters =
['numero','groupe','datetime','id_contact','surname','name','email_address','phone_number',
'statut_de_la_piste','proprietaire_de_la_piste','mc_form_id','lead_creation_date','hcp_erp_number',
'hearing_center','hcp_postal_code','hcp_phone_number','hcp_address','hcp_city','hcp_email_address','type']

contactsCallCenters = []
i = 0
for row in contacts:
    if row[15] == '05*****031':
        contactsCallCenters.append(row)
try:
    callCenterCsv = open(destination + '/Call Centers.csv', 'w', encoding='latin-1',
newline='')
    ecrire = csv.writer(callCenterCsv, delimiter=';')
    ecrire.writerow(colonnesCallCenters)
    for ligneContact in contactsCallCenters:
        ecrire.writerow(ligneContact)
except :
    print("Erreur try 6")
    print(" ")
    if jeton == False:
        print("Rien à faire!")
        print(" ")
print("Programme terminé")

```

```

Microsoft Windows [version 10.0.19042.985]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\administrateur>py mission.py

Connexion TLS établie ...
Création des fichiers ...

Programme terminé

C:\Users\administrateur>_

```

Figure 10