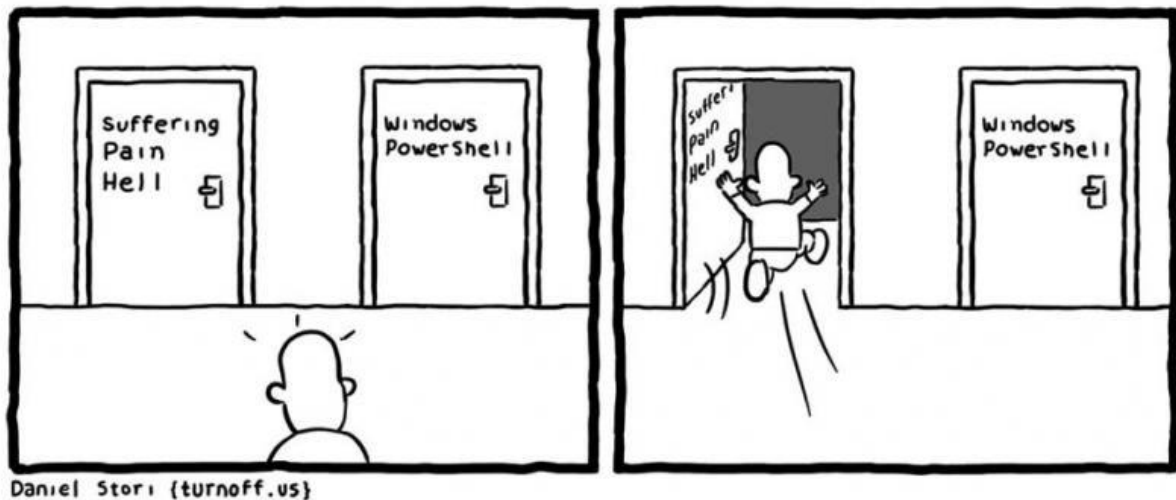


Gestion des utilisateurs dans l'AD de l'établissement



Daniel Stori (turnoff.us)

Figure 1 Image de titre

Sylvain Philipona – CIN4B
Ecole des Métiers Lausanne
Avenue de Valmont 30, 1010 Lausanne
89 heures
Roberto Ferrari
08.05.2023 – 07.06.2023

Table des matières

1	Analyse préliminaire	3
1.1	Introduction	3
1.2	Objectifs.....	4
1.3	Planification initiale	5
2	Analyse / Conception.....	6
2.1	Méthodologie de travail.....	6
2.2	Outil de versioning.....	7
2.3	Stratégie de test.....	8
2.3.1	Tests du script de création des comptes	8
2.3.2	Tests du script de modification des comptes	9
2.3.3	Tests du script de suppression des comptes	9
2.4	Risques techniques	10
2.5	Planification	10
2.6	Dossier de conception	11
2.6.1	Création des comptes.....	11
2.6.2	Modification des comptes	12
2.6.3	Suppression des comptes	13
3	Réalisation.....	15
3.1	Dossier de réalisation	15
3.1.1	Environnement.....	15
3.1.2	Version du projet.....	16
3.1.3	Normes de codage et de nommage.....	16
	Création des comptes.....	19
3.1.4	Modification des comptes	34
3.1.5	Suppression des comptes	43
3.2	Description des tests effectués.....	50
3.2.1	Tests du script de création des comptes	50
3.2.2	Tests du script de modification des comptes	52
3.2.3	Tests du script de suppression des comptes	53
3.3	Liste des documents fournis	54
3.4	Glossaire	54
4	Conclusion.....	55
5	Annexes.....	58
5.1	Résumé du rapport du TPI / version succincte de la documentation	58
5.2	Cahier des charges.....	59
5.3	Table d'illustrations	62
5.4	Sources – Bibliographie.....	64
5.5	Journal de travail	66
5.6	Planification détaillée	84
5.7	Archives du projet.....	96

1 Analyse préliminaire

1.1 Introduction

L'ETML, compte plus de 1200 étudiants et apprentis.

Chaque année on compte près de 500 nouveaux arrivants. En début d'année scolaire, il est nécessaire de gérer les comptes utilisateurs des élèves et apprentis. Pour cela il faut :

- Créer les comptes des nouveaux élèves et apprentis.
- Mettre à jour les comptes des élèves et des apprentis en cours de formation.
- Supprimer les comptes des élèves et des apprentis qui ont terminé leur cursus

Ces actions prennent beaucoup de temps, surtout si cela est fait à la main pour chaque compte utilisateur. C'est pourquoi il est nécessaire d'automatiser ces tâches au maximum afin de limiter l'action humaine et donc aussi limiter les pertes de temps et les erreurs humaines.

Le but de ce projet, est de proposer une solution de création / modification / suppression automatique des comptes utilisateur sur l'Active Directory (AD).

La technologie qui sera utilisée dans ce projet est PowerShell. La raison de ce choix réside dans le fait que PowerShell est un langage de Scripting natif à Windows et développé par Microsoft. De plus, y'a de nombreux composants installés par défaut permettant notamment d'interagir avec l'AD.

Le projet se déroule dans le cadre d'un Travail Pratique Individuel (TPI). L'environnement logiciel et matériel est le suivant :

- 1 PC standard de l'ETML
- Suite Office365
- Visual Studio Code
- Un serveur virtuel Windows 2022 avec un AD installé
- Un ordinateur virtuel Windows 10 dans le domaine de l'AD
- Un dépôt GitHub.

Les différents modules qui seront utiles au projet sont les suivants :

- Compétences en Scripting → Module 122 (PowerShell).
- Compétences en OS et Serveurs → Modules 305, 123 et 127 (MS Windows 10 et Server 2k19).
- Compétences en outils bureautiques → Module 302.
- Compétences en gestion de projets → Modules 306 et 431

1.2 Objectifs

Les objectifs techniques du projet définis dans le cahier des charges, sont les suivants :

1. Les scripts sont commentés.
2. Les comptes sont créés correctement, il est possible de se loguer, il est obligatoire de devoir changer son mot de passe.
3. Les dossiers personnels (home directory) sont créés avec les bons droits et sont montés au login du compte.
4. Les comptes sont modifiés correctement.
5. Les comptes sont supprimés / désactivés correctement.
6. Lors de la suppression du compte, le dossier personnel est supprimé correctement.
7. Les fichiers de sortie CSV contiennent toutes les informations nécessaires pour savoir ce qui a été créé, modifié, désactivé, supprimé.

1.3 Planification initiale

La planification initiale du projet a été élaborée en suivant la méthode des 6 pas. Le graphique ci-dessous représente le temps alloué pour chaque pas de la méthode. De plus les points de documentation et de visite des experts et du chef de projet ont été ajoutés. La planification détaillée comprenant en plus le découpage des tâches dans le temps est disponible en annexe. De plus, un comparatif entre la planification et la réalisation est disponible en conclusion de ce document. Le total d'heures planifiées est uniquement de 86, car je pense finir ce projet une demi-journée en avance. Ce qui correspond à 4 heures.

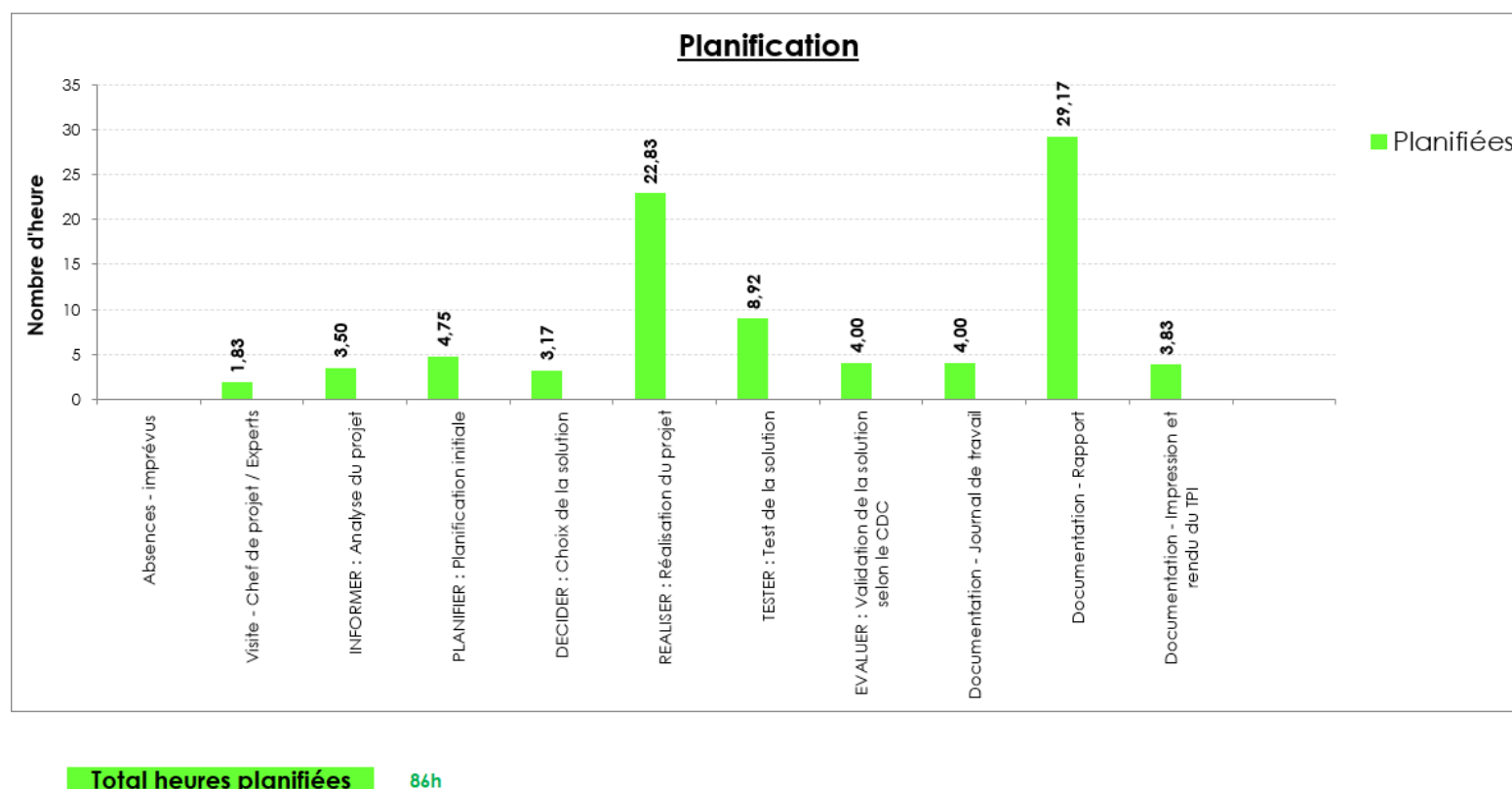


Figure 2 Planification initiale

2 Analyse / Conception

2.1 Méthodologie de travail

La méthodologie de projet que j'ai utilisé tout au long de ce projet se comme "La méthode des 6 pas/étapes". Cependant j'ai légèrement adapté cette méthode.

J'ai choisi cette méthode, car elle est relativement simple à mettre en place et en vue de l'envergure du projet et du nombre de personnes impliquées elle est efficace. De plus j'ai beaucoup utilisé cette méthode lors des différents projets tout au long de ma formation.

Voici en quoi consiste mon adaptation de cette méthode.



Figure 3 Schéma méthode des 6 pas



1. S'informer :

Cette étape est effectuée au tout début du projet. Elle consiste à analyser le projet dans son ensemble pour se faire une idée générale de ce qu'il y'a à faire. Cette étape permet de commencer le projet sur des bases solides et en étant conscient de l'ampleur du projet.



2. Planifier :

Une fois l'étape "informer" terminée, il faut planifier l'ensemble des tâches du projet dans le temps. Cette étape est cruciale au début du projet, car elle permet d'évaluer la durée prévue, pour sa réalisation. Avec une planification précise, il est simple d'avoir une vision d'ensemble du déroulement du projet.



Décider :

Cette étape ainsi que les suivantes sont effectuées pour chaque tâche planifiée. Durant l'étape "décider", il faut choisir la solution qui va être mise en place afin de réaliser au mieux la tâche. Cela implique d'évaluer les différentes options possibles, les risques et les bénéfices d'entre elles. Le choix de la meilleure solution permet de maximiser les chances de réussite de la tâche et du projet.



3. Réaliser :

L'étape de réalisation consiste basiquement en la réalisation de la tâche, en implémentant la solution choisie dans l'étape "décider". C'est à cette étape que le plus gros du travail du projet est réalisé. Cette étape est donc très importante, mais nécessite que les étapes précédentes de la méthode soient correctement effectuées.



4. Tester :

Une fois la réalisation de la tâche terminée, il faut effectuer des tests afin de vérifier le bon fonctionnement de la solution. Cette étape vise à vérifier si la solution répond aux critères de performance, de qualité et de conformité définis dans l'étape "décider". L'objectif est de détecter d'éventuels problèmes, bugs ou dysfonctionnements, afin de les corriger et d'améliorer la solution si nécessaire.



5. Evaluer :

Une fois la vérification du bon fonctionnement de la tâche effectuée, l'étape d'évaluation consiste à valider que la solution implémentée répond bien aux spécifications du cahier des charges. Les résultats obtenus sont évalués afin de vérifier qu'ils correspondent bien aux exigences initiales.

2.2 Outil de versioning

L'outil de versioning que j'ai utilisé, est Git avec la plateforme GitHub.

Vous pouvez trouver le répertoire GitHub de ce projet en suivant le lien ci-dessous :
<https://github.com/SylvainPhilipona/TPI-AD-Users-Management>

Le répertoire a été organisé en deux branches. "Main" et "Dev". Toutes les fonctionnalités sont développées dans la branche "Dev", puis une fois fonctionnelles, la branche est "merged" dans la branche "Main". Le projet est sauvegardé au moins une fois par jour.

2.3 Stratégie de test

La liste des tests à effectuer durant le projet pour valider son bon fonctionnement est décrite dans les trois tableaux ci-dessous. Chaque tableau correspond aux tests à effectuer dans un script. Tous les tests pouvant être automatisés facilement avec des tests unitaires le seront. Le reste est testé à la main. Les données de tests sont un mélange de données réels d'utilisateurs ainsi que des données fictives et erronées afin de tester les limitations des scripts.

2.3.1 Tests du script de création des comptes

Test	Impact en cas d'échec
Une recherche est effectuée pour vérifier si l'utilisateur existe déjà dans l'AD.	<ul style="list-style-type: none">• Cela peut entraîner une réinitialisation non voulue d'un compte utilisateur.
Le nom d'ouverture de session est généré avec les 3 premiers caractères du prénom + les 8 premiers caractères du nom de famille. De plus les diacritiques, espèces et apostrophes sont retirés.	<ul style="list-style-type: none">• Compromission du compte.• Impossibilité d'utilisation du compte.
Le nom d'ouverture de session proposé est unique, dans le cas contraire un autre est généré.	<ul style="list-style-type: none">• Risques de doublons d'identifiants• Impossibilité de connexion
Un mot de passe aléatoire devant être changé à l'ouverture de session est généré.	<ul style="list-style-type: none">• L'utilisateur ne peut pas changer de mot de passe.• Risques de sécurité.
Un script d'ouverture de session est généré selon la profession de l'utilisateur.	<ul style="list-style-type: none">• Accès inappropriés.• Difficultés de gestion
Une description est générée selon la classe de l'utilisateur.	<ul style="list-style-type: none">• Gestion des comptes plus compliquée.
L'utilisateur est créé dans la bonne OU.	<ul style="list-style-type: none">• Gestion des utilisateurs plus compliquée.• Problèmes de sécurité liés aux GPO.
Le dossier personnel est créé avec les bons droits d'accès.	<ul style="list-style-type: none">• Impossibilité d'avoir un dossier personnel.• Risques de compromission des données personnelles.
L'utilisateur est membre des groupes spécifiés dans le CSV.	<ul style="list-style-type: none">• Impossibilité d'utilisation du compte.
Un fichier CSV contenant les actions effectuées est généré.	<ul style="list-style-type: none">• Impossibilité d'obtenir un retour des actions effectuées.

2.3.2 Tests du script de modification des comptes

Test	Impact en cas d'échec
Une recherche par le nom d'ouverture de session est effectuée pour vérifier si l'utilisateur déjà dans l'AD.	<ul style="list-style-type: none">• Impossible de vérifier si un compte existe.• Peut générer des erreurs dans le script.
Le compte est modifié selon les informations du CSV.	<ul style="list-style-type: none">• Impossible de modifier les comptes.
Un fichier CSV contenant les modifications effectuées est généré.	<ul style="list-style-type: none">• Impossibilité d'obtenir un retour des actions effectuées.

2.3.3 Tests du script de suppression des comptes

Test	Impact en cas d'échec
Une recherche par le nom d'ouverture de session est effectuée pour vérifier si l'utilisateur existe dans l'AD.	<ul style="list-style-type: none">• Impossible de vérifier si un compte existe.• Peut générer des erreurs dans le script.
Il est possible de choisir de désactiver ou supprimer un compte.	<ul style="list-style-type: none">• Impossible d'utiliser le script.
Un compte est supprimé uniquement s'il est désactivé.	<ul style="list-style-type: none">• Perte de données.• Difficulté de suivi.
Le dossier personnel est supprimé lors de la suppression du compte.	<ul style="list-style-type: none">• Utilisation inutile d'espace disque.• Risques de fuite de données personnelles.
Un fichier CSV contenant les modifications et suppressions est généré.	<ul style="list-style-type: none">• Impossibilité d'obtenir un retour des actions effectuées.

2.4 Risques techniques

Le risque principal de ce projet est lié au remplissage des fichiers CSV par les utilisateurs de ce projet. Les données fournies peuvent être mauvaises voir incohérentes. Afin de limiter au maximum ce risque, de multiples tests sur les données fournies sont effectués.

2.5 Planification

Pour ce projet, la méthodologie de projet qui sera utilisée est la méthode des 6 pas. Ce projet se déroule en 5 semaines. Du lundi 8 mai 2023 à 8h au mercredi 7 juin 2023 à 16h35. Cela correspond à un total de 89 heures.

Les semaines durent 23h50m, et sont découpées de la manière suivante :

- Lundi -> 6h20
- Mercredi -> 7h10
- Jeudi -> 3h10
- Vendredi -> 7h10

De plus, les jours suivants n'ont pas été comptés dans les journées de travail :

- Jeudi 18 mai -> Ascension
- Vendredi 19 mai -> Pont de l'ascension
- Lundi 29 mai -> Pentecôte
- Jeudi 1^{er} juin -> Examen ECG

2.6 Dossier de conception

Les différents points (Création, modification et suppression) sont détaillés ci-dessous.

2.6.1 Création des comptes

Voici comment la création des comptes va se dérouler.

- La première étape consiste au remplissage des données des apprentis dans un fichier CSV. Ces données sont par exemple : Prénom, Nom, Classe... Une fois ce fichier rempli, l'utilisateur exécute le script de création des comptes.
- Une fois le script exécuté, toutes les données du fichier CSV vont être parcourues apprenti par apprenti. Le script va tester si le compte existe déjà. Si c'est le cas, le script passe à l'apprenti suivant. Dans le cas contraire, la phase de création du compte commence.
- Le script va passer par plusieurs étapes afin de créer le compte :
 1. Génération de l'identifiant de connexion. Suppression des diacritiques dans cet identifiant.
 2. Vérifie si l'identifiant existe déjà. Dans ce cas l'étape 1 est réexécutée afin de générer un nouvel identifiant. Dans le cas contraire, l'identifiant est gardé.
 3. Génération d'un mot de passe aléatoire. Génération d'une description renseignant la classe de l'apprenti.
 4. Création d'un dossier personnel (Home Directory) sur le serveur. Création d'un script d'ouverture de session qui va connecter au PC le dossier personnel de l'apprenti.
 5. Le compte est créé avec les paramètres générés dans les étapes précédentes et avec les données fournis dans les CSV. Les groupes l'apprenti est ensuite ajouté dans les différents groupes AD.
- Une fois ces étapes terminées, un fichier CSV est généré avec les informations concernant les comptes nouvellement créés.

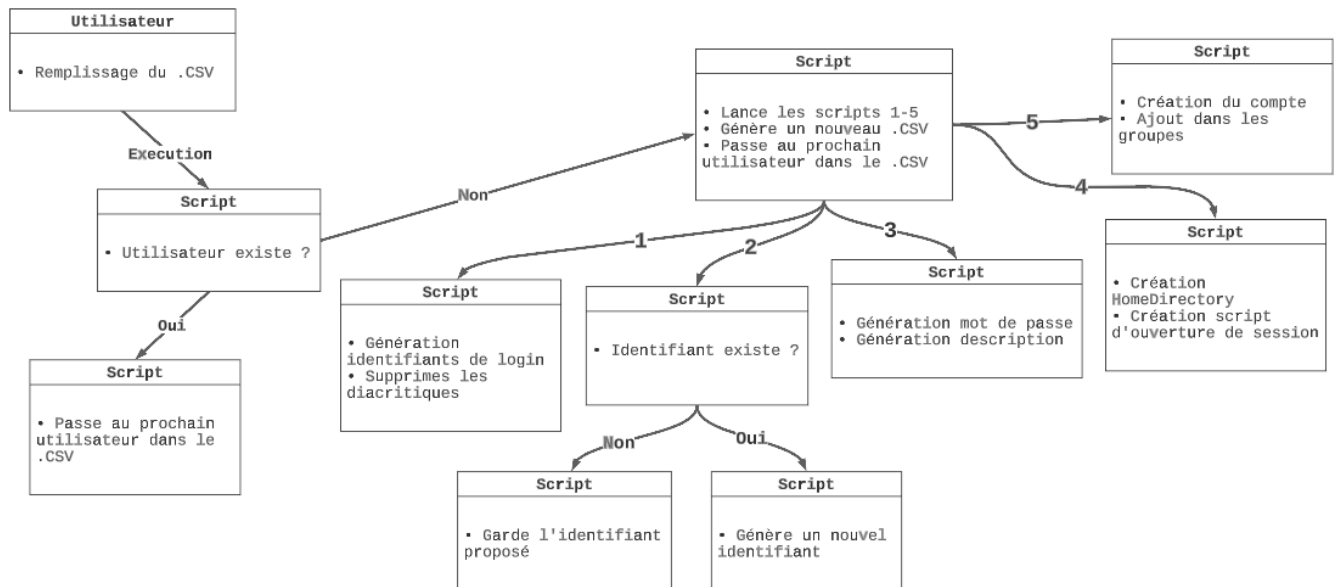


Figure 4 Schéma du processus de création des comptes

2.6.2 Modification des comptes

Voici comment la modification des comptes va se dérouler.

- La première étape consiste au remplissage des données des apprentis dans un fichier CSV. Ces données sont par exemple : Prénom, Nom, Classe... Une fois ce fichier rempli, l'utilisateur exécute le script de modification des comptes.
- Une fois le script exécuté, toutes les données du fichier CSV vont être parcourues apprenti par apprenti. Le script va tester si le compte existe. Si c'est le cas, la phase de modification du compte commence. Dans le cas contraire, le script passe à l'apprenti suivant.
- Pour la modification, le script va commencer par comparer les nouvelles informations fournies dans le CSV, et les comparer avec les informations actuelles des comptes. Toutes les informations qui sont différentes sont modifiées dans le compte. Ensuite un nouveau fichier CSV est généré avec les modifications effectuées.

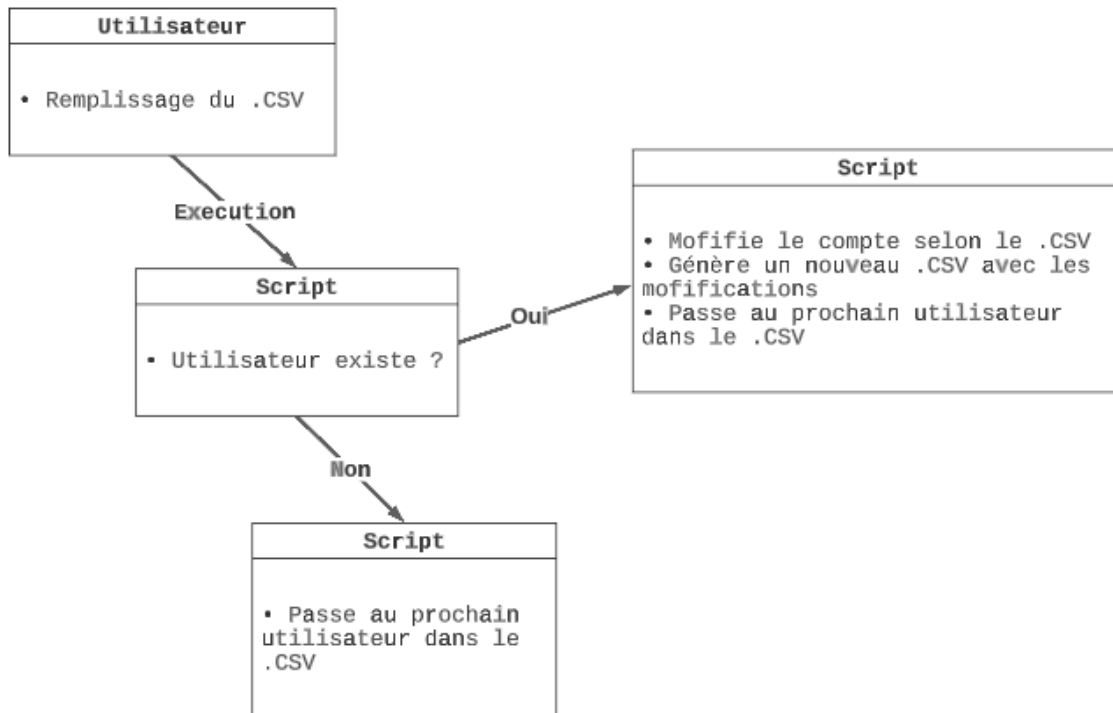


Figure 5 Schéma du processus de modification des comptes

2.6.3 Suppression des comptes

Voici comment la suppression des comptes va se dérouler.

- La première étape consiste au remplissage des données des apprentis dans un fichier CSV. Ces données sont par exemple : Prénom, Nom, Classe... Une fois ce fichier remplis, l'utilisateur exécute le script de suppression des comptes.
- Une fois le script exécuté, toutes les données du fichier CSV vont être parcourues apprenti par apprenti. Le script va tester si le compte existe. Si c'est le cas, la phase de suppression du compte commence. Dans le cas contraire, le script passe à l'apprenti suivant.
- Pour la suppression, le script va commencer par vérifier si le compte doit être désactivé ou bien supprimé. S'il doit être désactivé, le compte est simplement désactivé. Dans le cas d'une suppression, le script vérifie si le compte a été désactivé au préalable, et ensuite supprime le répertoire personnel puis enfin supprime le compte. Si le compte n'est pas désactivé, le script passe à l'utilisateur suivant. Ensuite un nouveau fichier CSV est généré avec les actions effectuées.

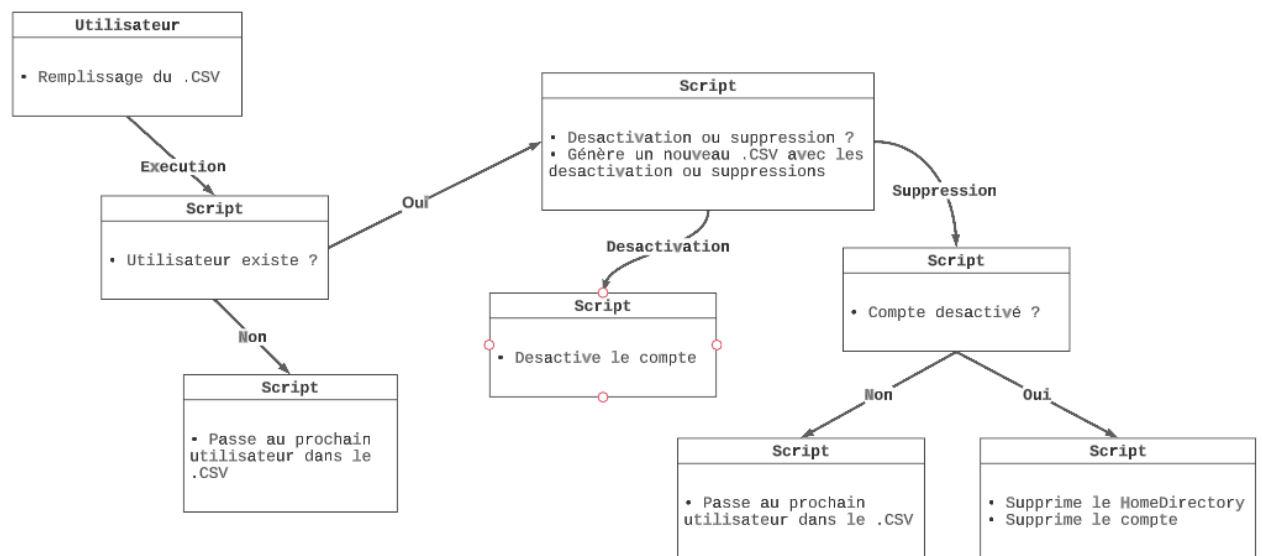


Figure 6 Schéma du processus de suppression des comptes

3 Réalisation

3.1 Dossier de réalisation

3.1.1 Environnement

Pour ce projet, j'avais à disposition un ordinateur standard de l'ETML. Les spécifications sont les suivantes :

- Windows 10 Education 21H2 64 Bits
- Intel i7-11700 2.50Ghz
- RAM 32 Gb 3200 MHz

L'environnement logiciel est le suivant :

- Visual Studio Code, Version 1.78.2
- GitHub Desktop, Version 3.2.3
- Oracle VirtualBox, Version 6.1
- Machine virtuelle, Version Windows Server 2022
- Machine virtuelle, Version Windows 10
- PowerShell, Version 5.1

Voici un schéma de l'environnement réseau utilisé durant ce projet.

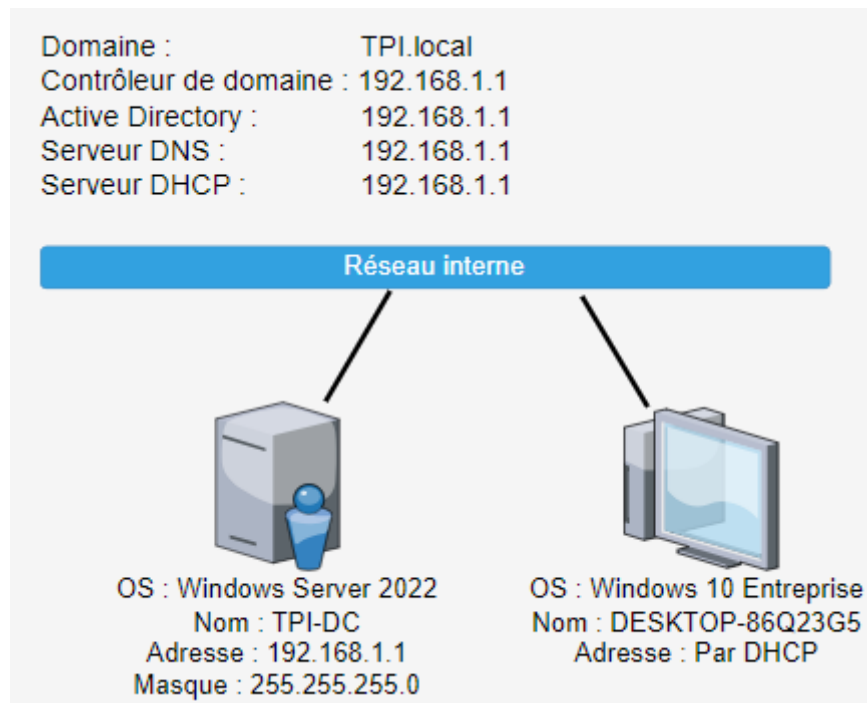


Figure 7 Schéma réseau de l'environnement

3.1.2 Version du projet

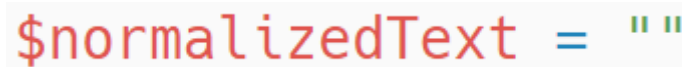
Ce projet est en version 1.0. Car je suis parti de 0 pour le réaliser.

3.1.3 Normes de codage et de nommage

Voici les normes de codage et de nommage utilisées tout au long de ce projet.

3.1.3.1 Variables et paramètres

Les noms de variables dans les scripts sont définis avec la norme "Lower Camel Case". C'est-à-dire que le nom de la variable débute par une lettre minuscule et les mots suivants commencent par une majuscule. Les noms de variables sont en anglais.



```
$normalizedText = ""
```

Figure 8 Format de nommage de variables

Les noms de paramètres dans les scripts sont définis avec la norme "Upper Camel Case". C'est-à-dire que le nom de la variable débute par une lettre majuscule et les mots suivants commencent aussi par une majuscule. De plus, le type du paramètre est forcé. Les noms de paramètres sont en anglais.

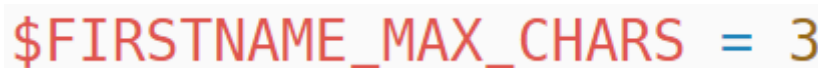


```
[string]$FolderPath
```

Figure 9 Format de nommage de paramètres

3.1.3.2 Constantes

Les noms de constantes dans les scripts sont définis sous le format suivant. Tous les mots de la constante sont en majuscule et séparés par un tiret bas. Les noms de constantes sont en anglais.



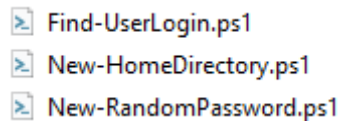
```
$FIRSTNAME_MAX_CHARS = 3
```

Figure 10 Format de nommage de constantes

3.1.3.3 Nom des scripts et tests unitaires

Les noms de scripts sont définis par les recommandations Microsoft de nommage des fonctions et scripts PowerShell. Cela consiste à choisir un "verbe" suivi d'un tiret et d'un nom. Le verbe doit faire partie de la liste approuvée par Microsoft et le nom doit spécifier ce qui va être affecté par le script. Les noms de scripts sont en anglais. Ces recommandations sont accessibles en suivant le lien ci-dessous :

<https://learn.microsoft.com/en-us/powershell/scripting/developer/cmdlet/approved-verbs-for-windows-powershell-commands?view=powershell-7.3>

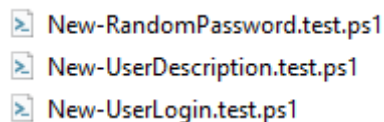


Find-UserLogin.ps1
New-HomeDirectory.ps1
New-RandomPassword.ps1

Figure 11 Format de nommage des scripts

Les noms des tests unitaires sont définis par les recommandations du Framework "Pester". Cela consiste à ajouter le mot ".test" entre le nom du script et l'extension du fichier. Ces recommandations sont accessibles en suivant le lien ci-dessous :

<https://pester.dev/docs/usage/file-placement-and-naming>



New-RandomPassword.test.ps1
New-UserDescription.test.ps1
New-UserLogin.test.ps1

Figure 12 Format de nommage des tests unitaires

3.1.3.4 Entêtes et commentaires

Les entêtes des scripts doivent être en anglais et contenir les informations suivantes :

1. Nom du script, auteur et date de création.
2. Date des modifications, auteur et raison de la modification.
3. Le titre du script.
4. Une description plus détaillée que le titre.
5. Une description de chaque paramètre.
6. Ce que le script retourne.
7. Un exemple de l'utilisation du script.
8. Les liens utilisés pour créer le script.

```

<#
.NOTES
*****
ETML 1
Name:  New-Example.ps1
Author: Sylvain Philipona
Date:  23.05.2023
*****
Modifications 2
Date : 24.05.2023
Author: Sylvain Philipona
Reason: New examples added
*****
.SYNOPSIS 3
Header example

.DESRIPTION 4
This script provides an exemple header used in powershell scripts

.PARAMETER Param1 5
The usfulness of this param

.OUTPUTS 6
- Output nothing

.EXAMPLE 7
.\New-Example.ps1

.LINK 8
https://example.com

#>

```

Figure 13 Format d'entête de scripts

Les commentaires dans les scripts sont pertinents et écrits en anglais.

```

# Split characters composed of several characters (é => e ´)
# Example : L'été de Raphaël => L ' e ´ t e ´ d e R a p h a e ´ l

```

Figure 14 Format de commentaires

Création des comptes

Le processus de création des comptes utilisateur commence tout d'abord par le remplissage d'un fichier CSV avec les données nécessaires. Ce fichier contient des informations telles que le prénom, le nom, l'adresse e-mail, la classe, la profession et les groupes dans lesquelles mettre l'utilisateur.

Ensuite, le script de création des comptes est exécuté. Pour chaque compte, le script vérifie les données du fichier CSV et effectue plusieurs opérations pour créer les comptes utilisateurs. Génération des identifiants de login et des mots de passe, génération de description, scripts de connexion, création des répertoire personnels et ajout dans les bons groupes. De plus, toutes les actions effectuées sont enregistrées dans un fichier CSV.

Dans cette section, nous détaillerons les différentes étapes du processus de création des comptes utilisateur, depuis le remplissage du fichier CSV jusqu'à la génération du fichier de journalisation.

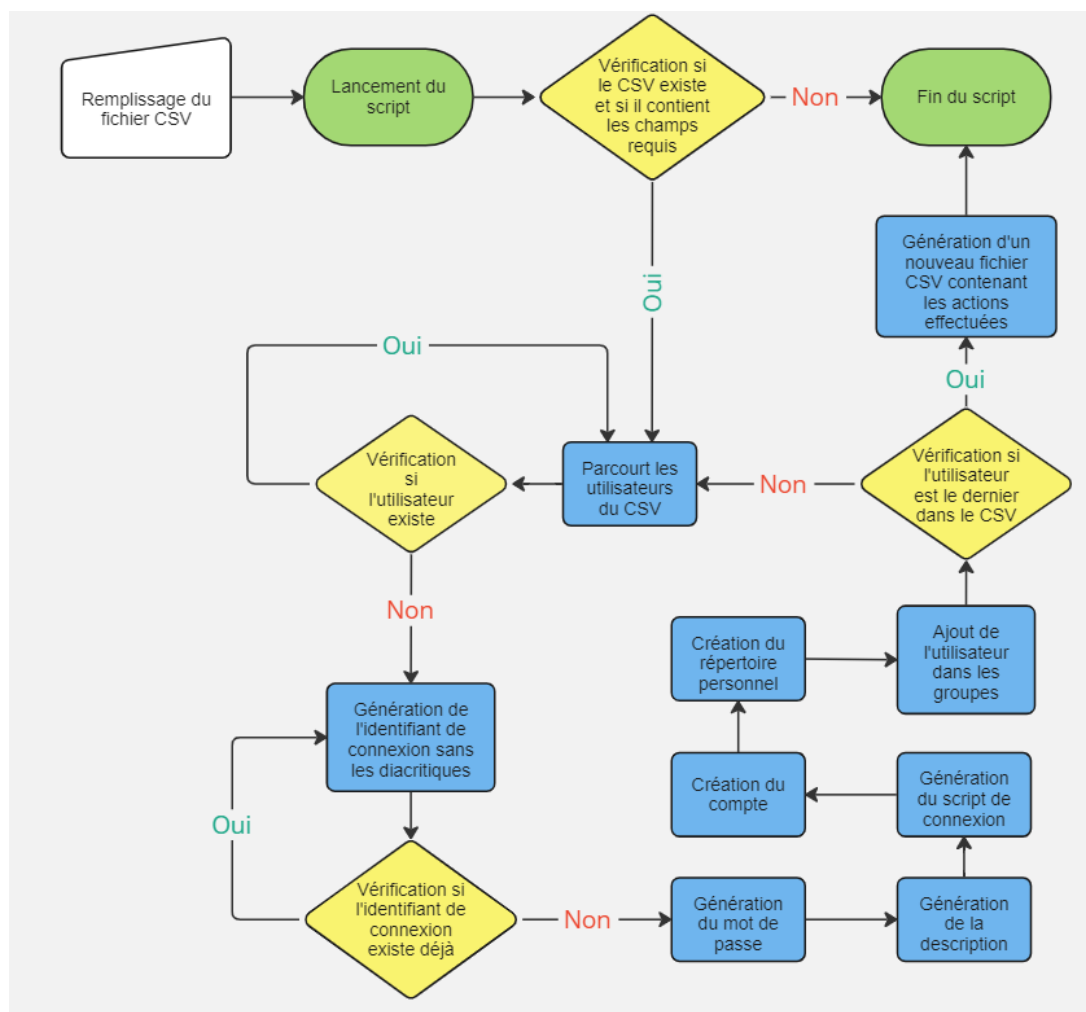


Figure 15 Diagramme de flux de la création des comptes

3.1.3.5 Remplissage du fichier CSV

Afin de pouvoir créer les comptes utilisateur, la première étape consiste à remplir un fichier CSV avec les données des futur comptes. Les données nécessaires à la création des comptes sont les suivantes :

- **Prénom**
- **Nom**
- **E-mail**
- **Classe**
- **Profession**
- **OptionsAD**

Prenom	Nom	E-mail	Classe	Profession	OptionsAD
Sylvain	Philipona	user1@eduvaud.ch	CIN2A	Informaticien CFC	APPL_SELECTLINE_RO%ELEVES%II
Joca	Bolli	user5@eduvaud.ch	CAM1	Automaticien CFC	APPL_SELECTLINE_RO%ELEVES%A
Nolan	Praz	user6@eduvaud.ch	CEM1	Menuisier CFC	APPL_SELECTLINE_RO%ELEVES%B
Alessandro	D'angélo	user7@eduvaud.ch	FLO1	Electronicien CFC	APPL_SELECTLINE_RO%ELEVES%E
Mihăescu	Negoită	user8@eduvaud.ch	FIN2	Informaticien CFC	APPL_SELECTLINE_RO%ELEVES%E

Figure 16 Exemple de données dans le CSV

Le champ "OptionsAD" doit contenir la liste des groupes AD à ajouter à l'utilisateur. Chaque groupe dans ce champ est séparé par un "%". Lors de la création des comptes la liste des groupes est obtenue en divisant la chaîne de caractères contenant les groupes aux endroits où il y'a le caractère "%". De plus, le préfix "GUS_ETML_" est ajouté au nom du groupe.

```
$groups = $user.OptionsAD.Split("%")
```

Figure 17 Division des groupes avec le caractère "%"

3.1.3.6 Exécution du script

Une fois le fichier CSV rempli avec les informations des comptes à créer, l'utilisateur peut exécuter le script. Lors de l'exécution ce celui-ci, il faut renseigner au script les paramètres suivants :

- **UsersCSV :**
Ce paramètre renseigne le chemin du fichier CSV contenant les données des comptes à créer. Ce paramètre doit obligatoirement être renseigné.
- **ActionsCSV :**
Ce paramètre renseigne le chemin du fichier CSV qui sera créé avec toutes les actions effectuées et les informations des nouveaux comptes (identifiant, mot de passe...). Ce paramètre doit obligatoirement être renseigné.

- **OUPath :**

Ce paramètre renseigne l'unité d'organisation (OU) où les comptes utilisateur vont être créés. Ce paramètre doit obligatoirement être renseigné.

- **CSVDelimiter**

Ce paramètre renseigne le caractère délimiteur des données du fichier CSV contenant les données des comptes à créer. Par défaut celui-ci est une virgule. Cependant certains fichiers ont en un autre comme un point-virgule par exemple. Ce paramètre est facultatif.

```
.\Start-UsersCreation.ps1
-UsersCSV "Z:\Administratif\01-creationUsers.csv"
-ActionsCSV "Z:\Administratif\02-createdUsers.csv"
-OUPath "OU=students,DC=tpi,DC=local"
-CSVDelimiter ";"
```

Figure 18 Exemple d'exécution du script de création avec des paramètres

3.1.3.7 Vérification des données du fichier CSV

Une fois le script de création exécuté, le fichier CSV et ses données sont vérifiés pour garantir le bon fonctionnement du programme. Les tests effectués sont les suivants :

1. **Vérification si le fichier existe et qu'il est dans le bon format**

Le chemin du fichier est testé afin de voir s'il existe. De plus le script vérifie si l'extension de fichier est bien ".csv". Si l'un des deux tests ne passe pas, un message d'erreur est affiché à l'utilisateur et l'exécution du script s'arrête.

```
# Check if the file exists and is in the .csv format
if(!(Test-Path -Path $UsersCSV -PathType Leaf) -or
    [IO.Path]::GetExtension($UsersCSV).ToLower() -ne ".csv"){
    Write-Host "The file '$UsersCSV' does not exist or is not in CSV format."
    exit
}
```

Figure 19 Vérification existence et format du CSV

2. **Vérification des en-têtes du fichier**

Le script teste si tous les champs d'entêtes de données nécessaires sont présents dans le fichier CSV. (Prénom, nom, E-mail, Classe...).

Pour ce faire, le script parcourt tous les champs nécessaires un par un, et vérifie si ce champ est présent dans l'entête. Si un des champs n'est pas présent dans l'entête, un message d'erreur est affiché à l'utilisateur et l'exécution du script s'arrête.

```
# Import the csv file and retrieve the csv headers
$users = Import-Csv $UsersCSV -Encoding UTF8 -Delimiter $CSVDelimiter
$csvHeaders = ($users | Get-Member -MemberType NoteProperty).Name

# Check if the csv contains all required headers
if(!(.\Test-ContainsHeaders.ps1 -CsvHeaders $csvHeaders -Headers $requiredHeaders)){
    Write-Host "The file '$UsersCSV' does not contain the required fields :
    $($requiredHeaders -Join(', '))."
    exit
}
```

Figure 20 Vérification des en-têtes du fichier

```
# Go through all required headers
foreach($header in $Headers){

    # Check if the CSV headers contains the required header
    if(!$CsvHeaders.Contains($header)){

        # If the CSV headers does not contains the required header
        return $false
    }
}

return $true
```

Figure 21 Fonction de vérification des en-têtes du fichier

3.1.3.8 Création des utilisateurs

Une fois ces vérifications effectuées, le script va parcourir tous les utilisateurs du fichier CSV un par un, puis les créer. Pour ce faire le script va effectuer les étapes décrites ci-dessous.

3.1.3.8.1 Vérification si l'utilisateur existe

Le script vérifie si l'utilisateur à créer existe déjà dans l'annuaire. Pour ce faire, une recherche sur le nom et le prénom de l'utilisateur est effectuée. Si aucune donnée n'est retournée, cela veut dire que le compte n'existe pas. Dans le cas contraire, l'utilisateur est ignoré et le script passe au prochain.

```
$adUser = (Get-Aduser -Filter {GivenName -eq $FirstName -and sn -eq $LastName})
```

Figure 22 Recherche d'un utilisateur par son nom et prénom

3.1.3.8.2 Génération de l'identifiant de connexion

Le nom de l'identifiant de connexion doit être généré sous le format suivant :
Les trois premiers caractères du prénom ainsi que les (maximum) 8 premiers caractères du nom de famille. De plus, cet identifiant ne doit pas contenir de caractères spéciaux dit "Diacritiques". Pour ce faire, le script procède en plusieurs étapes.

1. Suppression des caractères diacritiques

```
# Remove diacritics
$firstNameNoDiacr = .\Remove-Diacritics.ps1 -Text $user.$tagPrenom
$lastNameNoDiacr = .\Remove-Diacritics.ps1 -Text $user.$tagNom
```

Figure 23 Suppression des caractères diacritiques

Les caractères diacritiques sont supprimés sur les champs nom et prénom de l'utilisateur. Pour ce faire, la forme de normalisation "FormD" (Canonical Decomposition) est utilisée. Cette forme de normalisation Unicode spécifie que les caractères doivent être décomposés en utilisant la suite de caractères la plus basique possible. Par exemple le caractère "é" sous la forme "FormD" devient "e ´".

Ensuite, toute la chaîne de caractère normalisée est parcourue caractère par caractère. Chacun est testé s'il est de type "nonspacing". S'il ne l'est pas le caractère est ajouté à la chaîne finale. Le type de caractère "nonspacing" indique qu'un caractère de ce type ne prend pas d'espace visuel. Il s'agit notamment de caractères accentuant qui modifient un caractère de base. Par exemple le caractère " ´ " composé avec un "e" va l'accentuer et donner "é" sans prendre d'espace en plus.

```
# Split characters composed of several characters (é => e ´)
# Example : L'été de Raphaël => L ' e ´ t e ´ d e R a p h a e `l
[System.Text.NormalizationForm]$normalizationForm = "FormD"

# Normalize the text
$Normalized = $Text.Normalize($normalizationForm)

# Iterates through all text in the format of a character array
foreach($char in $normalized.ToCharArray()){

    # Nonspacing character that indicates modifications of a base character
    # Check if the character isn't nonspacing
    if ([Globalization.CharUnicodeInfo]::GetUnicodeCategory($char) -ne
        [Globalization.UnicodeCategory]::NonSpacingMark)
    {
        # Add the nonspacing character to the result string
        $normalizedText += $char
    }
}
```

Figure 24 Fonction de suppression des caractères diacritiques

2. Génération de l'identifiant de connexion

L'identifiant de connexion est généré en prenant les trois premiers caractères du prénom ainsi que les (maximum) 8 premiers caractères du nom de famille.

```
# Generate login name
$samAccountName = .\New-UserLogin.ps1 -FirstName $firstNameNoDiacr -LastName $lastNameNoDiacr
```

Figure 25 Génération de l'identifiant de connexion

3. Régénération si l'identifiant de connexion existe déjà

Un identifiant de connexion peut se trouver être en doublon avec un autre notamment dans le cas où un deux utilisateurs aurait un prénom proche et le même nom de famille. Par exemple :

Max Rochat => maxrochat
Maximilien Rochat => maxrochat
Maxence Rochat => maxrochat

Si l'identifiant de connexion généré existe déjà, un autre est généré en rajoutant un chiffre à la fin. Ce chiffre est incrémenté tant que le nouvel identifiant existe déjà. Si nous reprenons l'exemple précédent, avec cette régénération cela donnerait :

Max Rochat => maxrochat
Maximilien Rochat => maxrochat1
Maxence Rochat => maxrochat2

```
# Check if the login name already exists
# Regenerate the login name with a number at the end of it, while the generated name already exists
$i = 1
while (.\Find-UserLogin.ps1 -Login $samAccountName) {
    $samAccountName = "$(.\New-UserLogin.ps1 -FirstName $firstNameNoDiacr -LastName $lastNameNoDiacr)$i"
    $i++
}
```

Figure 26 Régénération si l'identifiant de connexion existe déjà

3.1.3.8.3 Génération du mot de passe

Une fois la génération de l'identifiant terminée, il faut créer un mot de passe aléatoire pour l'utilisateur. Pour ce faire, le script procède en trois étapes.

1. Liste des caractères utilisables

Lister les caractères de cette manière permet d'effectuer plus simplement des actions plus spécifiques lors de la génération comme par exemple l'exclusion de caractères similaires ou bien forcer les caractères spéciaux.

```
$UPPER = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"  
$LOWER = "abcdefghijklmnopqrstuvwxyz"  
$NUMBERS = "0123456789"  
$SPECIALS = "!#$%&*-/=?@_ "  
$SIMILAR = "IlL100o"  
$all = $UPPER + $LOWER + $NUMBERS + $SPECIALS
```

Figure 27 Liste des caractères utilisables

2. Génération aléatoire d'un caractère

Un nombre aléatoire entre 0 et la longueur de la liste contenant tous les caractères utilisables moins 1 est généré. Ce nombre représente un index dans cette liste. En récupérant le caractère à cet index le script obtient un caractère aléatoire. Cette étape ainsi que la suivante sont effectuées autant de fois que la longueur de mot de passe requise.

```
# Generate a random number from 0 to the max index of the 'all' variable  
$randomIndex = Get-Random -Minimum 0 -Maximum ($all.Length-1)  
  
# Get the char from the generated index and add it to the password  
$randomChar = $all[$randomIndex]
```

Figure 28 Génération aléatoire d'un caractère

3. Exclusion des caractères similaires

Si lors de l'appel de la fonction le paramètre "ExcludeSimilar" a été spécifié, le mot de passe ne doit pas contenir de caractères similaires qui pourraient être confondus par l'utilisateur. Ces caractères sont par exemple les "i" majuscules et les "L" minuscules.

Si le caractère généré se trouve être dans la liste des similaires, le script va simplement passer à la prochaine itération sans ajouter le caractère généré dans le mot de passe final.

```
# Check if the exclude similar param is specified
if($ExcludeSimilar -and $SIMILAR.Contains($randomChar)){

    # Pass to the next iteration
    continue
}
```

Figure 29 Exclusion des caractères similaires

3.1.3.8.4 Génération de la description

Chaque utilisateur doit avoir une description qui mentionne la classe dans laquelle il est. Cette description est générée en remplaçant le mot "[classe]" dans la description par défaut par l'actuelle classe de l'utilisateur.

```
$KEYWORD = "[classe]"
$description = "Cet utilisateur est membre de la classe $KEYWORD"
$description.Replace($KEYWORD, $Classe)
```

Figure 30 Génération de la description

3.1.3.8.5 Génération du script de connexion

A l'ETML, Il existe des scripts de connexion propres à chaque sections / professions. Selon la profession de l'utilisateur, le bon script de connexion doit lui être attribué. Ceci est fait en deux étapes.

1. Dictionnaire des professions

Un dictionnaire contenant les différentes professions ainsi que leurs scripts respectifs est déclaré. De cette manière il est simple d'attribuer un script selon la profession de l'utilisateur.

```
# The truth table for the professions and the scripts
$professionsScripts = @{
    "Automaticien"      = "AM.BAT"
    "Ebeniste"          = "BOIS.BAT"
    "Menuisier"         = "BOIS.BAT"
    "Electronicien"     = "ELO.BAT"
    "Informaticien"     = "INF.BAT"
    "Mechatronicien"    = "AUTO.BAT"
    "Polymecanicien"    = "PM.BAT"
    "Pre-apprentissage" = "PAPP.BAT"
    "Technicien"        = "ES.BAT"
    "MP-TASV"           = "TH.BAT"
    "MAD"               = "TH.BAT"
    "Technisceniste"    = "TECHNI.BAT"
}
```

Figure 31 Dictionnaire des professions

2. Attribution du script

Tous les champs du dictionnaire sont parcourus afin de trouver la profession de l'utilisateur dans le dictionnaire. Dès que celle-ci est trouvée, le script correspondant est retourné. Dans le cas où la profession ne serait pas retrouvée, la valeur null est retournée.

```
# Go through the whole truth table
foreach($script in $professionsScripts.GetEnumerator()){

    # Check if the profession in params match the element in the truth table
    if($Profession -like "*$($script.Key)*"){

        # Return the script
        return $script.Value
    }
}
```

Figure 32 Attribution du script

3.1.3.8.6 Création du compte

Une fois toutes les étapes précédentes effectuées, le compte utilisateur est prêt à être créé et activé. Pour ce faire le script procède en deux étapes.

1. Définition des données utilisateur

Un dictionnaire est créé contenant toutes les données nécessaires à la création de l'utilisateur.

```
# Set the proprieties of the new user
$newUserProps = @{
    displayName           = "$($user.$tagPrenom) $($user.$tagNom)"
    name                  = "$($user.$tagPrenom) $($user.$tagNom)"
    GivenName             = $user.$tagPrenom
    Surname                = $user.$tagNom
    samAccountName         = $samAccountName
    userPrincipalName      = $user.$tagEmail
    EmailAddress           = $user.$tagEmail
    AccountPassword        = ConvertTo-SecureString $password -AsPlainText -Force
    Description            = $description
    Path                   = $OUPath
    scriptPath             = $logonScript
}
```

Figure 33 Définition des données utilisateur

2. Création et activation du compte

Le compte est créé avec les données définies dans le dictionnaire. Ces données sont fournies avec l'opérateur "@" qui permet de passer les propriétés du dictionnaire en tant qu'arguments. Ensuite, la propriété "ChangePasswordAtLogon" est définie sur le compte utilisateur forçant le changement de mot de passe lors de la première connexion. Cette propriété doit être définie après la création du compte en raison d'un bug dans la fonction Microsoft de création de comptes AD. Pour finir le compte est activé.

```
# Create and enable the user  
New-ADUser @newUserProps  
Set-ADUser -Identity $samAccountName -ChangePasswordAtLogon $true  
Enable-ADAccount -Identity $samAccountName
```

Figure 34 Création et activation du compte

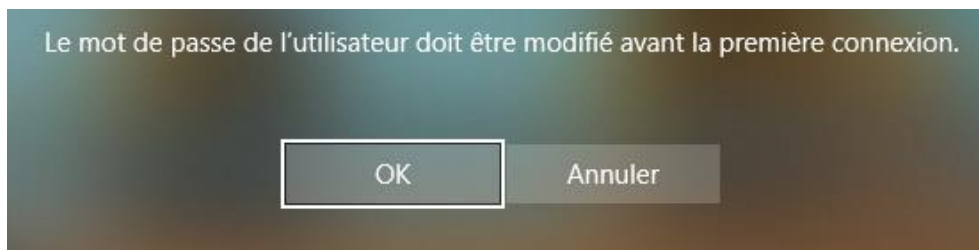


Figure 35 L'utilisateur doit changer de mot de passe

3.1.3.8.7 Création du répertoire personnel

Afin de créer le répertoire personnel de l'utilisateur, le script va passer par trois étapes.

1. Création du dossier principal

La première étape est de vérifier si le dossier principal contenant tous les dossiers personnels des utilisateurs existe. Si celui-ci n'existe pas, il est créé. L'héritage des droits est ensuite désactivé et les droits d'accès NTFS sont accordés en contrôle total au système et au propriétaire du dossier.

```
# Create the userhomes main folder if not exists and manage the access rights
if(!(Test-Path -Path $FolderPath)){

    # Create the folder
    New-Item -ItemType Directory -Path $FolderPath -Force -Confirm:$false | Out-Null

    # Disable inheritance
    $acl = Get-ACL -Path $FolderPath
    $acl.SetAccessRuleProtection($True, $True)
    Set-Acl -Path $FolderPath -AclObject $acl

    # Remove all access to the folder
    $acl = Get-ACL -Path $FolderPath
    $acl.Access | ForEach-Object {
        $acl.RemoveAccessRule($_) | Out-Null
    }

    # Add full control access to the folder owner and the system
    $acl.AddAccessRule((New-Object
        System.Security.AccessControl.FileSystemAccessRule($acl.Owner,"FullControl","Allow")))
    $acl.AddAccessRule((New-Object
        System.Security.AccessControl.FileSystemAccessRule("SYSTEM","FullControl","Allow")))
    Set-Acl -Path $FolderPath -AclObject $acl
}
```

Figure 36 Création du dossier principal

2. Création du répertoire personnel

Le script vérifie si le dossier personnel de l'utilisateur existe déjà. Si c'est le cas l'exécution continue sans recréer le dossier. Dans le cas contraire, le dossier personnel est créé. Ensuite, les droits d'accès NTFS sont accordés en uniquement en modification. De cette manière, l'utilisateur ne pourra pas modifier les droits d'accès à son dossier. Pour finir, le chemin d'accès réseau au dossier est retourné sous le format suivant :

"\\SRV-NAME\SHARE-NAME\username".

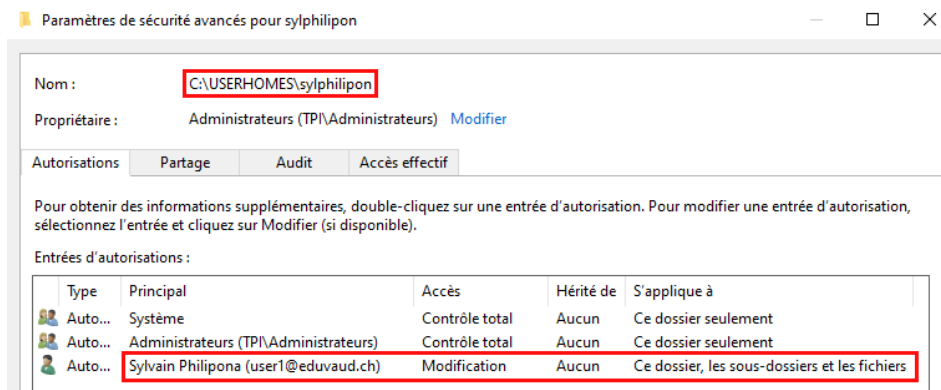


Figure 37 Droits d'accès au dossier

```
# Check if the user folder exists
if(!(Test-Path "$FolderPath\$Username")){
    # Create the user folder
    New-Item -Path "$FolderPath\$Username" -ItemType Directory -Force -Confirm:$false | Out-Null
}

# Add the access to the user if the user exists
# The access is in Modify
# The permissions are applied to both subfolders and files within the folder
$acl = Get-ACL -Path "$FolderPath\$Username"
$acl.AddAccessRule((New-Object System.Security.AccessControl.FileSystemAccessRule($Username, "Modify",
"ContainerInherit, ObjectInherit", "None", "Allow"))))
Set-Acl -Path "$FolderPath\$Username" -AclObject $acl

# Return the share path to the user home folder
return "\\$( $env:COMPUTERNAME )\$ShareName\$Username"
```

Figure 38 Création du répertoire personnel

3. Ajout du dossier dans le compte utilisateur

Le dossier personnel est ensuite ajouté dans les paramètres du compte utilisateur. Le dossier est associé à une lettre.

Figure 39 Chemin du dossier personnel

Cette lettre sera associée au lecteur réseau lors de l'accès au dossier personnel.

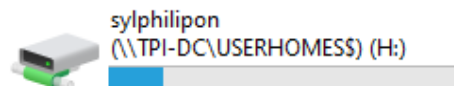


Figure 40 Lecteur réseau

3.1.3.8.8 Ajout dans les groupes

Une fois répertoire personnel de l'utilisateur créé, il faut maintenant l'ajouter dans les groupes. Le script reçoit en entrée la liste des groupes dans une chaîne de caractères séparés par un caractère "%".

```
APPL_SELECTLINE_RO%ELEVES%INF_Eleves%CDROMS_RO%WEBSERVER%
INF CINA2 Eleves%WEB MOBILITE USER%GRP-INTERNET%USERS STUDENTS
```

Figure 41 Exemples de chaîne de caractères de groupes

La liste des groupes est divisée aux endroits où il y a un "%", puis l'utilisateur est ajouté dans chaque groupe. De plus, le préfixe "GUS_ETML_" est ajouté au nom du groupe.

```
# Add the user to the groups
# Split the group field from the csv and add the groups prefix at the start of it
$groups = $user.$tagGroups.Split($GROUPS_SPLIT_CHAR) | ForEach-Object {
    $_.Replace($_, "$( $GROUPS_PREFIX)$($_)")
}
```

Figure 42 Obtention de la liste des groupes

```
$samAccountName | .\Add-UserGroups.ps1 -Groups $groups
```

Figure 43 Ajout dans les groupes

```
# Add the user to the group
Add-ADGroupMember -Identity $group -Members $adUser
```

Figure 44 Fonction d'ajout dans les groupes

3.1.3.9 Génération du CSV

Afin de journaliser ce qui se passe lors de la création des comptes, un fichier CSV contenant ce qui se passe est généré. Tout d'abord un dictionnaire contenant les titres des informations à journaliser est déclaré. Toutes les valeurs sont vides sauf l'utilisateur.

```
# Define the user actions data
# This data will be completed according to the script actions performed
$userActions = @{}
    User = "$($user.$tagPrenom) $($user.$tagNom)"
    Action = $null
    Login = $null
    Email = $null
    Password = $null
    LoginScript = $null
    HomeDirectory = $null
    Comments = $null
    Groups = $null
}
```

Figure 45 Déclaration du dictionnaire

Selon ce qui se passe durant la création du compte, les informations de journalisation sont remplies.

Ici, l'utilisateur existe déjà. L'action est définie sur "Failed" et le commentaire ajouté.

```
# The user already exists
$userActions["Action"] = "Failed"
$userActions["Comments"] = "The user '$($user.$tagPrenom) $($user.$tagNom)' already exists."
$allActions += [PSCustomObject]$userActions
continue
```

Figure 46 Erreur utilisateur existe déjà

Ici, il y'a eu une erreur lors de la création de l'utilisateur. L'action est définie sur "Failed", le commentaire ajouté ainsi que l'identifiant de connexion et le mot de passe.

```
# An error occurred during the account creation process
$userActions["Action"] = "Failed"
$userActions["Login"] = $samAccountName
$userActions["password"] = $password
$userActions["Comments"] = $_
$allActions += [PSCustomObject]$userActions
continue
```

Figure 47 Erreur lors de la création de l'utilisateur

Ici, le compte a été créé avec succès. Tous les champs sont donc renseignés. De plus, le dictionnaire est ajouté à la liste contenant toutes les actions des utilisateurs créés.

```
# Log the actions performed on the user
$userActions["Action"] = "Success"
$userActions["Login"] = $samAccountName
$userActions["Email"] = $user.$tagEmail
$userActions["Password"] = $password
$userActions["LoginScript"] = $logonScript
$userActions["HomeDirectory"] = $homeDir
$userActions["Comments"] = "The account '$samAccountName' was successfully created."
$userActions["Groups"] = ($groupsActions -join $CSVDelimiter)
$allActions += [PSCustomObject]$userActions
```

Figure 48 Compte créé avec succès

Une fois tous les comptes créés, le fichier csv est exporté avec les données des utilisateurs.

User	Action	Login	Email	Password	LoginScript	Comments
Joca TestNull	Failed					The user 'Joca TestNull' has empty fields.
Sylvain Philipona	Success	sylphilipon	user1@eduvaud.ch	UYxMtg#hzW!q	INF.BAT	The account 'sylphilipon' was successfully created.
Sylvain Philipone	Success	sylphilipon1	user3@eduvaud.ch	W29X3&MR5e#c	INF.BAT	The account 'sylphilipon1' was successfully created.
Sylvain Philiponu	Success	sylphilipon2	user4@eduvaud.ch	ynA*SiKWb!Ak	INF.BAT	The account 'sylphilipon2' was successfully created.
Joca Bolli	Success	jocbolli	user5@eduvaud.ch	bjhb/D@ZKeDW	AM.BAT	The account 'jocbolli' was successfully created.
Nolan Praz	Success	nolpraz	user6@eduvaud.ch	*FC6s44XRkPw	BOIS.BAT	The account 'nolpraz' was successfully created.
Alessandro D'angelo	Success	aledangelo	user7@eduvaud.ch	bUZb2x-sxfxQ	ELO.BAT	The account 'aledangelo' was successfully created.
Mihănescu Negoită	Success	mihnegoita	user8@eduvaud.ch	iGuHxk@*zrgX	INF.BAT	The account 'mihnegoita' was successfully created.

Figure 49 Données CSV exportés

3.1.4 Modification des comptes

Le processus de modification des comptes utilisateur commence tout d'abord par le remplissage d'un fichier CSV avec les données nécessaires. Ce fichier contient des informations telles que le nom d'ouverture de session, l'e-mail, la classe, la profession et les groupes dans lesquelles mettre l'utilisateur.

Ensuite, le script de modification des comptes est exécuté. Pour chaque compte, le script vérifie les données du fichier CSV et effectue plusieurs opérations afin de modifier les comptes utilisateurs. Génération des descriptions, scripts de connexion, modification de l'utilisateur et ajout dans les bons groupes. De plus, toutes les actions effectuées sont enregistrées dans un fichier CSV.

Dans cette section, nous détaillerons les différentes étapes du processus de modification des comptes utilisateur, depuis le remplissage du fichier CSV jusqu'à la génération du fichier de journalisation.

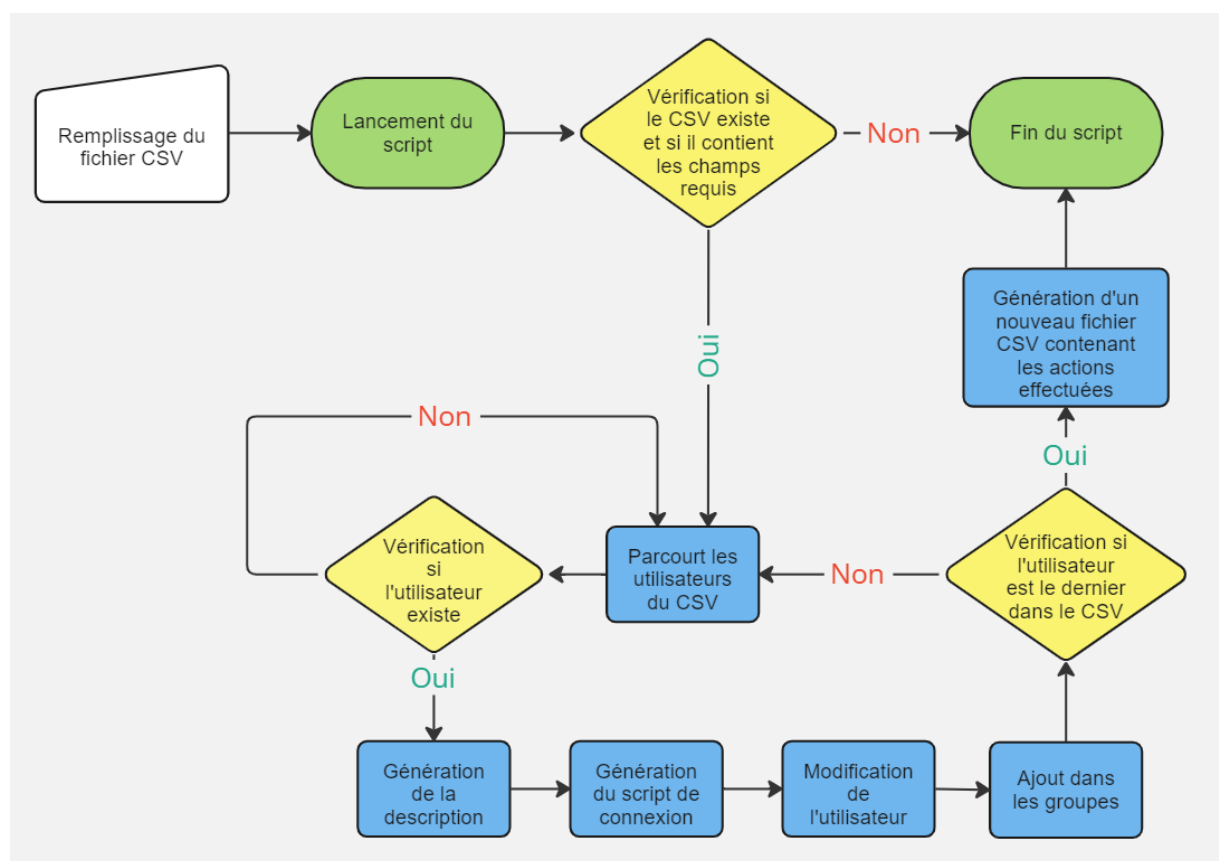


Figure 50 Diagramme de flux de la modification des comptes

3.1.4.1 Remplissage du fichier CSV

Afin de pouvoir modifier les comptes utilisateur, la première étape consiste à remplir un fichier CSV avec les nouvelles données des comptes. Les données nécessaires à la modification des comptes sont les suivantes :

- **Identifiant de connexion**
- **Nouvel E-mail**
- **Nouvelle Classe**
- **Nouvelle Profession**
- **Nouvelles OptionsAD**

Login	E-mail	Classe	Profession	OptionsAD
sylphilipon	sylphilipona@eduvaud.ch	CIN4B	Informaticien CFC	APPL_SELECTLINE_RO%ELEVES%EI
sylphilipon1	ni@gmail.com	FIN1	Informaticien CFC	APPL_SELECTLINE_RO%ELEVES%EI
jocbolli	jocbolli@gmail.com	CIN1A	Informaticien CFC	APPL_SELECTLINE_RO%ELEVES%EI
nolpraz	dodo@coca.ch	CID2B	Informaticien CFC	APPL_SELECTLINE_RO%ELEVES%EI
aledangelo	contact@aless.ch	CID3B	Informaticien CFC	APPL_SELECTLINE_RO%ELEVES%EI

Figure 51 Exemple de données dans le CSV

Le champ "OptionsAD" doit contenir la liste des groupes AD à ajouter à l'utilisateur. Chaque groupe dans ce champ est séparé par un "%". Lors de la création des comptes la liste des groupes est obtenue en divisant la chaîne de caractères contenant les groupes aux endroits où il y'a le caractère "%". De plus, le préfix "GUS_ETML_" est ajouté au nom du groupe.

```
$groups = $user.OptionsAD.Split( "%")
```

Figure 52 Division des groupes avec le caractère "%"

3.1.4.2 Exécution du script

Une fois le fichier CSV rempli avec les informations des comptes à modifier, l'utilisateur peut exécuter le script. Lors de l'exécution ce celui-ci, il faut renseigner au script les paramètres suivants :

- **UsersCSV :**
Ce paramètre renseigne le chemin du fichier CSV contenant les données des comptes à modifier. Ce paramètre doit obligatoirement être renseigné.
- **ActionsCSV :**
Ce paramètre renseigne le chemin du fichier CSV qui sera créé avec toutes les actions effectuées. Ce paramètre doit obligatoirement être renseigné.

- **CSVDelimiter**

Ce paramètre renseigne le caractère délimiteur des données du fichier CSV contenant les données des comptes à modifier. Par défaut celui-ci est une virgule. Cependant certains fichiers ont en un autre comme un point-virgule par exemple. Ce paramètre est facultatif.

```
.\Start-UsersModification.ps1
-UsersCSV "Z:\CSV\01-modificationUsers.csv"
-ActionsCSV "Z:\CSV\02-modifiedUsers.csv"
-CSVDelimiter ";"
```

Figure 53 Exemple d'exécution du script de modification avec des paramètres

3.1.4.3 Vérification des données du fichier CSV

Une fois le script de modification exécuté, le fichier CSV et ses données sont vérifiés pour garantir le bon fonctionnement du programme. Les tests effectués sont les suivants :

3. Vérification si le fichier existe et qu'il est dans le bon format

Le chemin du fichier est testé afin de voir s'il existe. De plus le script vérifie si l'extension de fichier est bien ".csv". Si l'un des deux tests ne passe pas, un message d'erreur est affiché à l'utilisateur et l'exécution du script s'arrête.

```
# Check if the file exists and is in the .csv format
if(!(Test-Path -Path $UsersCSV -PathType Leaf) -or
    [IO.Path]::GetExtension($UsersCSV).ToLower() -ne ".csv"){
    Write-Host "The file '$UsersCSV' does not exist or is not in CSV format."
    exit
}
```

Figure 54 Vérification existence et format du CSV

4. Vérification des en-têtes du fichier

Le script teste si tous les champs d'entêtes de données nécessaires sont présents dans le fichier CSV. (Identifiant de connexion, E-mail, Classe...).

Pour ce faire, le script parcourt tous les champs nécessaires un par un, et vérifie si ce champ est présent dans l'entête. Si un des champs n'est pas présent dans l'entête, un message d'erreur est affiché à l'utilisateur et l'exécution du script s'arrête.

```
# Import the csv file and retrieve the csv headers
$users = Import-Csv $UsersCSV -Encoding UTF8 -Delimiter $CSVDelimiter
$csvHeaders = ($users | Get-Member -MemberType NoteProperty).Name

# Check if the csv contains all required headers
if(!(.\Test-ContainsHeaders.ps1 -CsvHeaders $csvHeaders -Headers $requiredHeaders)){
    Write-Host "The file '$UsersCSV' does not contain the required fields :
    $($requiredHeaders -Join(', '))."
    exit
}
```

Figure 55 Vérification des en-têtes du fichier

```
# Go through all required headers
foreach($header in $Headers){

    # Check if the CSV headers contains the required header
    if(!$CsvHeaders.Contains($header)){

        # If the CSV headers does not contains the required header
        return $false
    }
}

return $true
```

Figure 56 Fonction de vérification des en-têtes du fichier

3.1.4.4 Modification des utilisateurs

Une fois ces vérifications effectuées, le script va parcourir tous les utilisateurs du fichier CSV un par un, puis les modifier. Pour ce faire le script va effectuer les étapes décrites ci-dessous.

3.1.4.4.1 Vérification si l'utilisateur existe

Le script vérifie si l'utilisateur à modifier existe déjà dans l'annuaire. Pour ce faire, une recherche sur le nom d'ouverture de session de l'utilisateur est effectuée. Si une donnée est retournée, cela veut dire que le compte existe. Dans le cas contraire, l'utilisateur est ignoré et le script passe au prochain.

```
$adUser = Get-ADUser -Filter {samAccountName -eq $samAccountName}
```

Figure 57 Recherche d'un utilisateur

3.1.4.4.2 Génération de la description

Chaque utilisateur doit avoir une description qui mentionne la classe dans laquelle il est. Cette description est générée en remplaçant le mot "[classe]" dans la description par défaut par l'actuelle classe de l'utilisateur.

```
$KEYWORD = "[classe]"
$description = "Cet utilisateur est membre de la classe $KEYWORD"
$description.Replace($KEYWORD, $Classe)
```

Figure 58 Génération de la description

3.1.4.4.3 Génération du script de connexion

A l'ETML, Il existe des scripts de connexion propres à chaque sections / professions. Selon la profession de l'utilisateur, le bon script de connexion doit lui être attribué. Ceci est fait en deux étapes.

1. Dictionnaire des professions

Un dictionnaire contenant les différentes professions ainsi que leurs scripts respectifs est déclaré. De cette manière il est simple d'attribuer un script selon la profession de l'utilisateur.

```
# The truth table for the professions and the scripts
$professionsScripts = @{
    "Automaticien"      = "AM.BAT"
    "Ebeniste"          = "BOIS.BAT"
    "Menuisier"         = "BOIS.BAT"
    "Electronicien"     = "ELO.BAT"
    "Informaticien"     = "INF.BAT"
    "Mecatronicien"     = "AUTO.BAT"
    "Polymecanicien"    = "PM.BAT"
    "Pre-apprentissage" = "PAPP.BAT"
    "Technicien"        = "ES.BAT"
    "MP-TASV"           = "TH.BAT"
    "MAD"               = "TH.BAT"
    "Technisceniste"    = "TECHNI.BAT"
}
```

Figure 59 Dictionnaire des professions

2. Attribution du script

Tous les champs du dictionnaire sont parcourus afin de trouver la profession de l'utilisateur dans le dictionnaire. Dès que celle-ci est trouvée, le script correspondant est retourné. Dans le cas où la profession ne serait pas retrouvée, la valeur null est retournée.

```
# Go through the whole truth table
foreach($script in $professionsScripts.GetEnumerator()){

    # Check if the profession in params match the element in the truth table
    if($Profession -like "*${$script.Key}*"){

        # Return the script
        return $script.Value
    }
}
```

Figure 60 Attribution du script

3.1.4.4.4 Modification de l'utilisateur

Une fois toutes les étapes précédentes effectuées, le compte utilisateur est prêt à être modifié. Pour ce faire le script procède en deux étapes.

1. Définition des données utilisateur

Un dictionnaire est créé contenant toutes les données nécessaires à la modification de l'utilisateur.

```
# Set the new proprieties of the user
$newUserProps = @{
    Identity           = $samAccountName
    EmailAddress       = $user.$tagEmail
    userPrincipalName  = $user.$tagEmail
    Description        = $description
    scriptPath         = $loginScript
}
```

Figure 61 Définition des données utilisateur

2. Modification du compte

Le compte est modifié avec les données définies dans le dictionnaire. Ces données sont fournies avec l'opérateur "@" qui permet de passer les propriétés du dictionnaire en tant qu'arguments.

```
# Edit the user
Set-ADUser @newUserProps
```

Figure 62 Modification du compte

3.1.4.4.5 Ajout dans les groupes

Une fois compte utilisateur modifié, il faut maintenant l'ajouter dans les groupes. Le script reçoit en entrée la liste des groupes dans une chaîne de caractères séparés par un caractère "%".

```
APPL_SELECTLINE_RO%ELEVES%INF_Eleves%CDROMS_RO%WEBSERVER%  
INF CINA2 Eleves%WEB MOBILITE USER%GRP-INTERNET%USERS STUDENTS
```

Figure 63 Exemples de chaîne de caractères de groupes

Il faut d'abord retirer l'utilisateur de tous ses groupes.

```
# Remove the user from all groups  
$aduser.MemberOf | ForEach-Object {  
    $_ | Remove-ADGroupMember -Members $adUser -Confirm:$false  
}
```

Figure 64 Retirer l'utilisateur des groupes

La liste des groupes est divisée aux endroits où il y a un "%", puis l'utilisateur est ajouté dans chaque groupe. De plus, le préfix "GUS_ETML_" est ajouté au nom du groupe.

```
# Add the user to the groups  
# Split the group field from the csv and add the groups prefix at the start of it  
$groups = $user.$tagGroups.Split($GROUPS_SPLIT_CHAR) | ForEach-Object {  
    $_.Replace($_, "$($GROUPS_PREFIX)$($_)")  
}
```

Figure 65 Obtention de la liste des groupes

```
$samAccountName | .\Add-UserGroups.ps1 -Groups $groups
```

Figure 66 Ajout dans les groupes

```
# Add the user to the group  
Add-ADGroupMember -Identity $group -Members $adUser
```

Figure 67 Fonction d'ajout dans les groupes

3.1.4.5 Génération du CSV

Afin de journaliser ce qui se passe lors de la modification des comptes, un fichier CSV contenant ce qui se passe est généré. Tout d'abord un dictionnaire contenant les titres des informations à journaliser est déclaré. Toutes les valeurs sont vides sauf l'utilisateur.

```
# Define the user actions data
# This data will be completed according to the script actions performed
$userActions = @{
    User = $samAccountName
    Action = $null
    Email = $null
    Classe = $null
    LoginScript = $null
    Comments = $null
    Groups = $null
}
```

Figure 68 Déclaration du dictionnaire

Selon ce qui se passe durant la modification du compte, les informations de journalisation sont remplies.

Ici, l'utilisateur n'existe pas. L'action est définie sur "Failed" et le commentaire ajouté.

```
# The user doesn't exists
$userActions["Action"] = "Failed"
$userActions["Comments"] = "The user '$samAccountName' doesn't exists."
$allActions += [PSCustomObject]$userActions
continue
```

Figure 69 Erreur utilisateur n'existe pas

Ici, il y'a eu une erreur lors de la modification de l'utilisateur. L'action est définie sur "Failed", le commentaire ajouté.

```
# An error occurred during the account modification process
$userActions["Action"] = "Failed"
$userActions["Comments"] = $_
$allActions += [PSCustomObject]$userActions
continue
```

Figure 70 Erreur lors de la modification de l'utilisateur

Ici, le compte a été modifié avec succès. Tous les champs sont donc renseignés. De plus, le dictionnaire est ajouté à la liste contenant toutes les actions des utilisateurs créés.

```
# Log the actions performed on the user
$userActions["Action"] = "Success"
$userActions["Email"] = $user.$tagEmail
$userActions["Classe"] = $user.$tagClasse
$userActions["LoginScript"] = $loginScript
$userActions["Comments"] = "The account '$samAccountName' was successfully modified."
$userActions["Groups"] = ($groupsActions -join $CSVDelimiter)
$allActions += [PSCustomObject]$userActions
```

Figure 71 Compte modifié avec succès

Une fois tous les comptes modifiés, le fichier csv est exporté avec les données des utilisateurs.

User	Action	Email	Classe	LoginScript	Comments	Groups
sylphilipon	Success	sylphilipona@eduvaud.ch	CIN2B	INF.BAT	The account 'sylphilipon' was successfully modified.	GUS_ETML_AP...
sylphilipon1	Success	ni@gmail.com	FIN1	INF.BAT	The account 'sylphilipon1' was successfully modified.	GUS_ETML_AP...
	Failed				The user '' has empty fields.	
jocbolli	Success	jocbolli@gmail.com	CIN1A	INF.BAT	The account 'jocbolli' was successfully modified.	GUS_ETML_AP...
nolpraz	Success	dodo@coca.ch	CID2B	INF.BAT	The account 'nolpraz' was successfully modified.	GUS_ETML_AP...
aledangelo	Success	contact@aless.ch	CID2B	INF.BAT	The account 'aledangelo' was successfully modified.	GUS_ETML_AP...
test	Failed				The user 'test' doesn't exists.	

Figure 72 Données CSV exportés

3.1.5 Suppression des comptes

Le processus de suppression des comptes utilisateur commence tout d'abord par le remplissage d'un fichier CSV avec les données nécessaires. Ce fichier contient des informations telles que le nom d'ouverture de session, une indication si le compte doit être supprimé ou bien désactivé.

Ensuite, le script de suppression des comptes est exécuté. Pour chaque compte, le script vérifie les données du fichier CSV et effectue plusieurs opérations afin de supprimer les comptes utilisateurs. Désactivation, suppression du répertoire personnel et suppression du compte utilisateur. De plus, toutes les actions effectuées sont enregistrées dans un fichier CSV.

Dans cette section, nous détaillerons les différentes étapes du processus de suppression des comptes utilisateur, depuis le remplissage du fichier CSV jusqu'à la génération du fichier de journalisation.

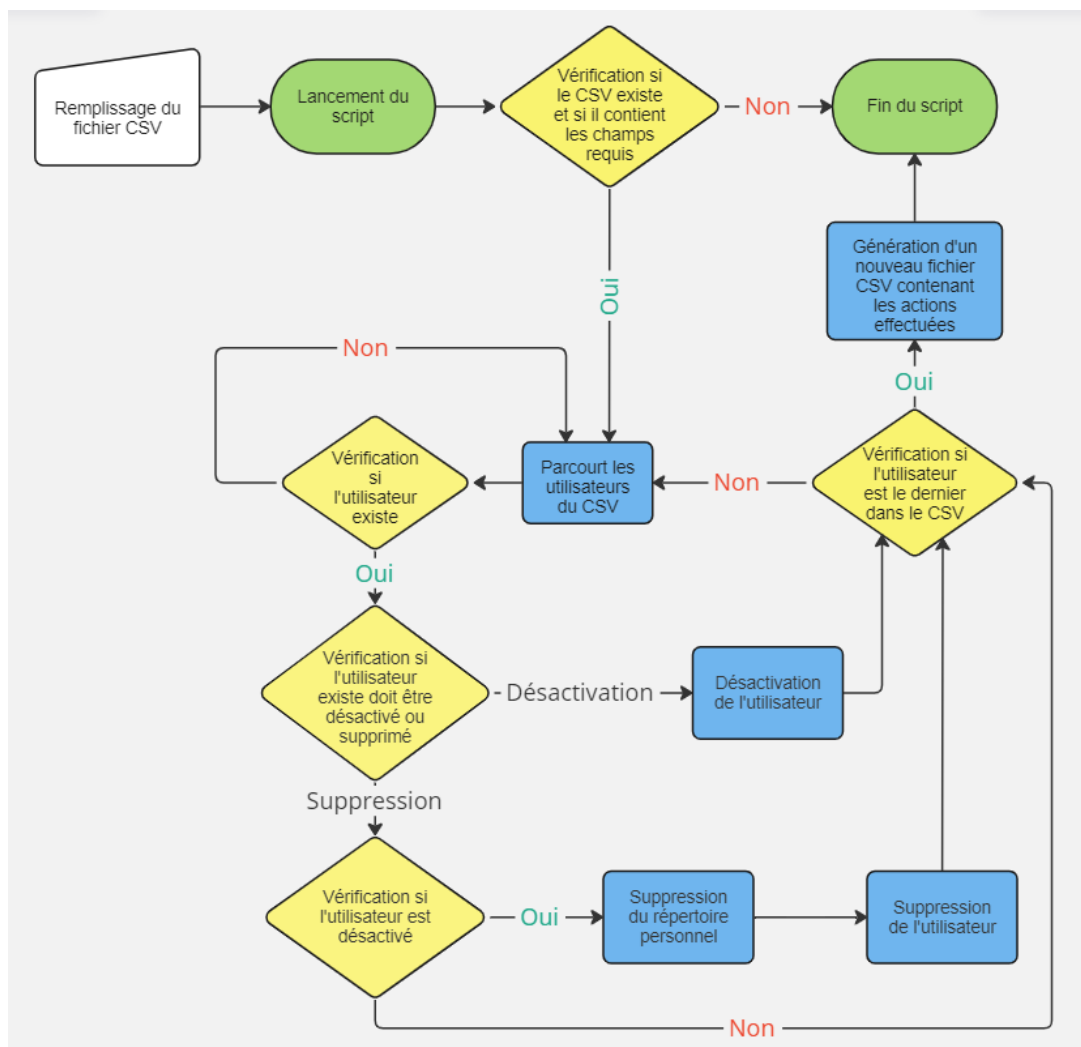


Figure 73 Diagramme de flux de la modification des comptes

3.1.5.1 Remplissage du fichier CSV

Afin de pouvoir modifier les comptes utilisateur, la première étape consiste à remplir un fichier CSV avec les nouvelles données des comptes. Les données nécessaires à la modification des comptes sont les suivantes :

- **Identifiant de connexion**
- **Désactivation**
- **Suppression**

Login	Disable	Delete
sylphilipon		True
sylphilipon1		True
sylphilipon2	True	
jocbolli	True	
nolpraz	True	
aledangelo		True
test	True	

Figure 74 Exemple de données dans le CSV

Les champs "Disable et "Delete" doivent contenir la valeur "True" pour effectuer l'action, ou bien n'importe quelle valeur pour ne pas l'effectuer. Il absolument faut un champ en "True" et l'autre avec une autre valeur. Dans le cas où les deux champs ont la valeur "True", le script priorise la désactivation. Un compte doit obligatoirement être supprimé avant de pouvoir être supprimé.

3.1.5.2 Exécution du script

Une fois le fichier CSV remplis avec les informations des comptes à supprimer, l'utilisateur peut exécuter le script. Lors de l'exécution ce celui-ci, il faut renseigner au script les paramètres suivants :

- **UsersCSV :**
Ce paramètre renseigne le chemin du fichier CSV contenant les données des comptes à supprimer. Ce paramètre doit obligatoirement être renseigné.
- **ActionsCSV :**
Ce paramètre renseigne le chemin du fichier CSV qui sera créé avec toutes les actions effectuées. Ce paramètre doit obligatoirement être renseigné.
- **CSVDelimiter**
Ce paramètre renseigne le caractère délimiteur des données du fichier CSV contenant les données des comptes à modifier. Par défaut celui-ci est une virgule. Cependant certains fichiers ont en un autre comme un point-virgule par exemple. Ce paramètre est facultatif.

```
.\Start-UsersDeletion.ps1  
-UsersCSV "Z:\CSV\01-deletionUsers.csv"  
-ActionsCSV "Z:\CSV\02-deletedUsers.csv"  
-CSVDelimiter ";"
```

Figure 75 Exemple d'exécution du script de suppression avec des paramètres

3.1.5.3 Vérification des données du fichier CSV

Une fois le script de modification exécuté, le fichier CSV et ses données sont vérifiés pour garantir le bon fonctionnement du programme. Les tests effectués sont les suivants :

1. Vérification si le fichier existe et qu'il est dans le bon format

Le chemin du fichier est testé afin de voir s'il existe. De plus le script vérifie si l'extension de fichier est bien ".csv". Si l'un des deux tests ne passe pas, un message d'erreur est affiché à l'utilisateur et l'exécution du script s'arrête.

```
# Check if the file exists and is in the .csv format
if(!(Test-Path -Path $UsersCSV -PathType Leaf) -or
    [IO.Path]::GetExtension($UsersCSV).ToLower() -ne ".csv"){
    Write-Host "The file '$UsersCSV' does not exist or is not in CSV format."
    exit
}
```

Figure 76 Vérification existence et format du CSV

2. Vérification des en-têtes du fichier

Le script teste si tous les champs d'entêtes de données nécessaires sont présents dans le fichier CSV. (Identifiant de connexion, désactivation...). Pour ce faire, le script parcourt tous les champs nécessaires un par un, et vérifie si ce champ est présent dans l'entête. Si un des champs n'est pas présent dans l'entête, un message d'erreur est affiché à l'utilisateur et l'exécution du script s'arrête.

```
# Import the csv file and retrieve the csv headers
$users = Import-Csv $UsersCSV -Encoding UTF8 -Delimiter $CSVDelimiter
$csvHeaders = ($users | Get-Member -MemberType NoteProperty).Name

# Check if the csv contains all required headers
if(!(.\Test-ContainsHeaders.ps1 -CsvHeaders $csvHeaders -Headers $requiredHeaders)){
    Write-Host "The file '$UsersCSV' does not contain the required fields :
    $($requiredHeaders -Join(', '))."
    exit
}
```

Figure 77 Vérification des en-têtes du fichier

```
# Go through all required headers
foreach($header in $Headers){

    # Check if the CSV headers contains the required header
    if(!$CsvHeaders.Contains($header)){

        # If the CSV headers does not contains the required header
        return $false
    }
}

return $true
```

Figure 78 Fonction de vérification des en-têtes du fichier

3.1.5.4 Modification ou suppression des utilisateurs

3.1.5.4.1 Vérification si l'utilisateur existe

Une fois ces vérifications effectuées, le script va parcourir tous les utilisateurs du fichier CSV un par un. Le script vérifie si l'utilisateur à supprimer existe déjà dans l'annuaire. Pour ce faire, une recherche sur le nom d'ouverture de session de l'utilisateur est effectuée. Si une donnée est retournée, cela veut dire que le compte existe. Dans le cas contraire, l'utilisateur est ignoré et le script passe au prochain.

```
$adUser = Get-ADUser -Filter {samAccountName -eq $samAccountName}
```

Figure 79 Recherche d'un utilisateur

3.1.5.4.2 Désactivation de l'utilisateur

Afin de savoir si le compte doit être désactivé ou bien supprimé, le script va vérifier si la valeur du champ "Disable" est "True". Dans ce cas, le script va simplement désactiver l'utilisateur. Dans le cas contraire le script va passer à l'étape de suppression du compte.

```
# Disable the ad user  
$adUser | Disable-ADAccount -Confirm:$false
```

Figure 80 Désactivation d'un compte utilisateur

3.1.5.4.3 Suppression de l'utilisateur

Le script va tout d'abord vérifier si la valeur du champ "Delete" est "True". Dans ce cas, le processus de suppression du compte va commencer. Dans le cas contraire, le script passe au prochain utilisateur dans le CSV.

3.1.5.4.3.1 Vérification si l'utilisateur est désactivé

Le script va tester si l'utilisateur est désactivé dans l'annuaire. Dans ce cas les étapes de suppression continuent. Dans le cas contraire, le script passe au prochain utilisateur dans le CSV.

```
# Check if the user is disabled  
# Every users has to be disabled before deleted  
if($adUser.Enabled -eq $False){
```

Figure 81 Vérification si l'utilisateur est désactivé

3.1.5.4.3.2 Suppression du répertoire personnel

Le script commence par récupérer le chemin du dossier personnel de l'utilisateur, puis supprime récursivement le dossier ainsi que ses sous dossiers et fichier. Ensuite, les champs "HomeDirectory" et "HomeDrive" sont définis avec une valeur nulle.

```
# Remove the home directory of the user
# Unmapping the home directory in the ad user
$adUser.HomeDirectory | Remove-Item -Recurse -Force -Confirm:$false
$adUser | Set-ADUser -HomeDirectory $null -HomeDrive $null
```

Figure 82 Suppression du répertoire personnel

3.1.5.4.3.3 Suppression de l'utilisateur

L'utilisateur est supprimé de l'annuaire.

```
# Delete the ad user
$adUser | Remove-ADUser -Confirm:$false
```

Figure 83 Suppression de l'utilisateur

3.1.5.4.4 Génération du CSV

Afin de journaliser ce qui se passe lors de la suppression des comptes, un fichier CSV contenant ce qui se passe est généré. Tout d'abord un dictionnaire contenant les titres des informations à journaliser est déclaré. Toutes les valeurs sont vides sauf l'utilisateur.

```
# Define the user actions data
# This data will be completed according to the script actions performed
$userActions = @{
    User = $userLogin
    Action = $null
    Comments = $null
}
```

Figure 84 Déclaration du dictionnaire

Selon ce qui se passe durant la suppression du compte, les informations de journalisation sont remplies.

Ici, l'utilisateur n'existe pas. L'action est définie sur "Failed" et le commentaire ajouté.

```
# The user doesn't exists
$userActions["Action"] = "Failed"
$userActions["Comments"] = "The user '$userLogin' doesn't exists."
$allActions += [PSCustomObject]$userActions
continue
```

Figure 85 Erreur utilisateur n'existe pas

Ici, l'utilisateur doit être supprimé mais n'est pas désactivé, L'action est définie sur "Failed" et le commentaire ajouté.

```
$userActions["Action"] = "Failed"
$userActions["Comments"] = "No actions were performed on the user '$userLogin'.
The user has to be disabled before deleted."
```

Figure 86 Erreur utilisateur pas désactivé

Ici, il y'a eu un problème avec les données de désactivation et de suppression. Cela peut être causé si aucun des deux champs ne contient la valeur "True".

```
$userActions["Action"] = "Failed"
$userActions["Comments"] = "No actions were performed on the user '$userLogin'.
There is a contradiction between the 'Disable' and the 'Delete' values."
```

Figure 87 Erreur problèmes avec les champs "Disable" et "Delete"

Ici, l'utilisateur a bien été supprimé, cependant il n'avait pas de répertoire personnel.

```
$userActions["Action"] = "Success"
$userActions["Comments"] = "The user '$userLogin' successfully deleted,
but he didn't have a home directory."
```

Figure 88 Utilisateur supprimé mais il n'avait pas de répertoire personnel

Une fois tous les comptes supprimés ou désactivés, le fichier csv est exporté avec les données des utilisateurs.

User	Action	Comments
sylphilipon	Success	The user 'sylphilipon' and his home directory was successfully deleted.
sylphilipon1	Success	The user 'sylphilipon1' and his home directory was successfully deleted.
sylphilipon2	Success	The user 'sylphilipon2' has been disabled.
jocbolli	Success	The user 'jocbolli' has been disabled.
nolpraz	Success	The user 'nolpraz' has been disabled.
aledangelo	Failed	No actions were performed on the user 'aledangelo'. The user has to be disabled before deleted.
test	Failed	The user 'test' doesn't exists.

3.2 Description des tests effectués

3.2.1 Tests du script de création des comptes

Test	Impact en cas d'échec	Résultat attendu	Résultat obtenu	Conclusion	Temps requis pour corriger
Une recherche est effectuée pour vérifier si l'utilisateur existe déjà dans l'AD.	<ul style="list-style-type: none"> • Cela peut entrainer une réinitialisation non voulue d'un compte utilisateur. 	<ul style="list-style-type: none"> • Un utilisateur existant est trouvé. Mais un non existant n'est pas trouvé. 	OK 100%	Rien à signaler	
Le nom d'ouverture de session est généré avec les 3 premiers caractères du prénom + les 8 premiers caractères du nom de famille. De plus les diacritiques, espèces et apostrophes sont retirés.	<ul style="list-style-type: none"> • Compromission du compte. • Impossibilité d'utilisation du compte. 	<ul style="list-style-type: none"> • Le nom d'ouverture est conçu avec les 3 premiers caractères du prénom + les 8 premiers caractères du nom. • Le nom d'ouverture ne contient pas de caractères diacritiques. 	OK 100%	Via test unitaire	
Le nom d'ouverture de session proposé est unique, dans le cas contraire un autre est généré.	<ul style="list-style-type: none"> • Risques de doublons d'identifiants • Impossibilité de connexion 	<ul style="list-style-type: none"> • Un nom d'ouverture existe uniquement une seul fois. 	OK 100%	Rien à signaler	
Génération d'un mot de passe aléatoire devant être changé à l'ouverture de session.	<ul style="list-style-type: none"> • L'utilisateur ne peut pas changer de mot de passe. • Risques de sécurité. 	<ul style="list-style-type: none"> • Le mot de passe est aléatoire et doit être changé à la première ouverture de session. 	OK 100%	Génération du mot de passe via test unitaire	

Test	Impact en cas d'échec	Résultat attendu	Résultat obtenu	Conclusion	Temps requis pour corriger
Un script d'ouverture de session est généré selon la profession de l'utilisateur.	<ul style="list-style-type: none"> • Accès inappropriés. • Difficultés de gestion 	<ul style="list-style-type: none"> • Le script d'ouverture de session est celui de la profession de l'utilisateur. 	OK 100%	Via test unitaire	
Une description est générée selon la classe de l'utilisateur.	<ul style="list-style-type: none"> • Gestion des comptes plus compliquée. 	<ul style="list-style-type: none"> • La description de l'utilisateur est générée avec sa classe. 	OK 100%	Via test unitaire	
L'utilisateur est créé dans la bonne OU.	<ul style="list-style-type: none"> • Gestion des utilisateurs plus compliquée. • Problèmes de sécurité liés aux GPO. 	<ul style="list-style-type: none"> • L'utilisateur est créé dans l'OU spécifiée par l'utilisateur. 	OK 100%	Rien à signaler	
Le dossier personnel est créé avec les bons droits d'accès.	<ul style="list-style-type: none"> • Impossibilité d'avoir un dossier personnel. 	<ul style="list-style-type: none"> • Le dossier personnel est créé avec les bons droits. • L'utilisateur y'a accès. 	OK 100%	Rien à signaler	
L'utilisateur est membre des groupes spécifiés dans le CSV.	<ul style="list-style-type: none"> • Impossibilité d'utilisation du compte. 	<ul style="list-style-type: none"> • L'utilisateur est membre des groupes du fichier CSV. 	OK 100%	Rien à signaler	
Un fichier CSV contenant les actions effectuées est généré.	<ul style="list-style-type: none"> • Impossibilité d'obtenir un retour des actions effectuées. 	<ul style="list-style-type: none"> • Toutes les actions effectuées sont écrites dans un nouveau fichier CSV. 	OK 100%	Rien à signaler	

3.2.2 Tests du script de modification des comptes

Test	Impact en cas d'échec	Résultat attendu	Résultat obtenu	Conclusion	Temps requis pour corriger
Une recherche par le nom d'ouverture de session est effectuée pour vérifier si l'utilisateur déjà dans l'AD.	<ul style="list-style-type: none">• Impossible de vérifier si un compte existe.• Peut générer des erreurs dans le script.	<ul style="list-style-type: none">• Un utilisateur existant est trouvé. Mais un non existant n'est pas trouvé.	OK 100%	Rien à signaler	
Le compte est modifié selon les informations du CSV.	<ul style="list-style-type: none">• Impossible de modifier les comptes.	<ul style="list-style-type: none">• Le compte est modifié avec les bonnes informations	OK 100%	Rien à signaler	
Un fichier CSV contenant les modifications effectuées est généré.	<ul style="list-style-type: none">• Impossibilité d'obtenir un retour des actions effectuées.	<ul style="list-style-type: none">• Toutes les actions effectuées sont écrites dans un nouveau fichier CSV.	OK 100%	Rien à signaler	

3.2.3 Tests du script de suppression des comptes

Test	Impact en cas d'échec	Résultat attendu	Résultat obtenu	Conclusion	Temps requis pour corriger
Une recherche par le nom d'ouverture de session est effectuée pour vérifier si l'utilisateur existe dans l'AD.	<ul style="list-style-type: none"> Impossible de vérifier si un compte existe. Peux générer des erreurs dans le script. 	<ul style="list-style-type: none"> Un utilisateur existant est trouvé. Mais un non existant n'est pas trouvé. 	OK 100%	Rien à signaler	
Il est possible de choisir de désactiver ou supprimer un compte.	<ul style="list-style-type: none"> Impossible d'utiliser le script. 	<ul style="list-style-type: none"> L'utilisateur peut choisir entre la suppression et la désactivation d'un compte. 	OK 100%	Rien à signaler	
Un compte est supprimé uniquement s'il est désactivé.	<ul style="list-style-type: none"> Perte de données. Difficulté de suivi. 	<ul style="list-style-type: none"> Si un compte n'est pas désactivé, il ne peut pas être supprimé. 	OK 100%	Rien à signaler	
Le dossier personnel est supprimé lors de la suppression du compte.	<ul style="list-style-type: none"> Utilisation inutile d'espace disque. Risques de fuite de données personnelles. 	<ul style="list-style-type: none"> Le dossier personnel et toutes ses informations sont supprimés. 	OK 100%	Rien à signaler	
Un fichier CSV contenant les modifications et suppressions est généré.	<ul style="list-style-type: none"> Impossibilité d'obtenir un retour des actions effectuées. 	<ul style="list-style-type: none"> Toutes les actions effectuées sont écrites dans un nouveau fichier CSV. 	OK 100%	Rien à signaler	

3.3 Liste des documents fournis

Document	Version
Le rapport de projet	1.0
Le code source du projet	1.0
Journal de travail	1.0
Planification détaillée	1.0

Les documents cités ci-dessus, excepté pour le rapport de projet, sont disponible dans la partie annexe de ce document.

3.4 Glossaire

Mot	Définition
Active Directory (AD)	Un service de gestion des utilisateurs, des ordinateurs et des ressources réseau dans un environnement Windows.
Annuaire	Une base de données organisée contenant des informations sur des personnes et des ressources.
CSV	Un format de fichier utilisé pour stocker des données sous forme de valeurs séparées par des virgules.
Diacritiques	Des signes ou des symboles ajoutés aux caractères pour indiquer des variations.
Domaine	Un groupe d'ordinateurs et d'utilisateurs partageant une configuration et une sécurité communes dans un réseau.
Droits NTFS	Les autorisations de fichiers et de dossiers utilisées par le système de fichiers de Windows pour gérer l'accès et les privilèges des utilisateurs et des groupes.
Fonction	Un bloc de code qui effectue une tâche spécifique et peut être appelé et réutilisé à plusieurs endroits dans un programme.
Forme de normalisation	Un processus d'organisation et de structuration de données.
Framework	Une structure de développement logiciel qui fournit des fonctionnalités et des outils pour faciliter la création d'applications.
GPO	Un outil utilisé pour gérer et appliquer des paramètres de configuration aux ordinateurs et aux utilisateurs dans un environnement Windows.

Mot	Définition
Itération	Le processus de répétition d'un bloc de code ou d'une série d'instructions dans le but d'effectuer une tâche plusieurs fois.
Merged	La fusion de deux éléments distincts en un seul.
Scripting	L'écriture et l'exécution de scripts pour automatiser des tâches ou effectuer des actions spécifiques.
Tests unitaires	Des tests réalisés pour vérifier le bon fonctionnement d'une unité de code ou d'un module spécifique dans un logiciel.
Unicode	Un standard de codage de caractères qui vise à représenter de manière universelle les caractères et symboles de toutes les écritures du monde.
Unité d'organisation (OU)	Une unité de gestion dans un Active Directory utilisée pour organiser les objets (utilisateurs, ordinateurs, groupes).
Valeur null ou nulle	Une valeur spéciale utilisée pour représenter l'absence de données ou une valeur non définie.

4 Conclusion

Pour conclure ce projet, je vais procéder en plusieurs étapes. Les objectifs, les difficultés particulières, la suite possible pour le projet, un bilan personnel et pour finir un bilan de planification.

Les objectifs :

Je vais aborder objectifs du cahier des charges et voir s'ils ont été atteints.

- **Les scripts sont commentés.**

Tous les scripts sont entièrement commentés. En-têtes + commentaires pertinents. Ce point est donc fonctionnel à 100%.

- **Les comptes sont créés correctement, il est possible de se loguer, il est obligatoire de devoir changer son mot de passe.**

Les comptes sont créés selon les informations dans le CSV. Il est possible de se connecter avec les comptes sur les ordinateurs membres du domaine. De plus, lors de la première connexion, les utilisateurs doivent changer leurs mots de passe. Ce point est donc fonctionnel à 100%.

- **Les dossiers personnels (home directory) sont créés avec les bons droits et sont montés au login du compte.**
Lors de la création des comptes, un dossier personnel par compte est créé avec les bons droits d'accès. Ces dossiers sont montés au login des comptes. Ce point est donc fonctionnel à 100%.
- **Les comptes sont modifiés correctement.**
Les comptes sont modifiés selon les informations dans le CSV. Ce point est donc fonctionnel à 100%.
- **Les comptes sont supprimés / désactivés correctement.**
Les comptes sont supprimés ou désactivés selon les informations dans le CSV. Ce point est donc fonctionnel à 100%.
- **Lors de la suppression du compte, le dossier personnel est supprimé correctement.**
Lors de la suppression des comptes, les dossiers personnels des utilisateurs sont supprimés, ainsi que leurs contenus. Ce point est donc fonctionnel à 100%.
- **Les fichiers de sortie CSV contiennent toutes les informations nécessaires pour savoir ce qui a été créé, modifié, désactivé, supprimé.**
Pour chaque scripts (Création, modification, suppression), un fichier CSV est généré avec toutes les informations nécessaires, ainsi que les actions effectuées. Ce point est donc fonctionnel à 100%.

Les difficultés particulières :

Il y'a eu très peu de difficultés dans ce projet. La grande majorité des tâches étaient relativement simples. Cependant un point qui a été un peu challengeant est la suppression des diacritiques. Ce point m'a demandé d'approfondir la manière dont fonctionne la gestion des caractères avec Unicode. Ce qui était très intéressant mais assez compliqué.

Les suites possibles du projet :

Le projet va être mis en production sur l'AD de l'ETML. L'utilisation de ce projet sera complètement libre. L'ETML pourra à se guise l'utiliser et le modifier. Le projet va être aussi disponible sur mon dépôt GitHub.

Bilan personnel :

J'ai particulièrement apprécié travailler sur ce projet. Aimant beaucoup automatiser des tâches, la réalisation de ce projet était très agréable. De plus j'ai pu consolider mes connaissances en PowerShell ainsi que dans la gestion de comptes dans un Active Directory. Ce sont des connaissances qui me seront utiles tout au long de ma carrière.

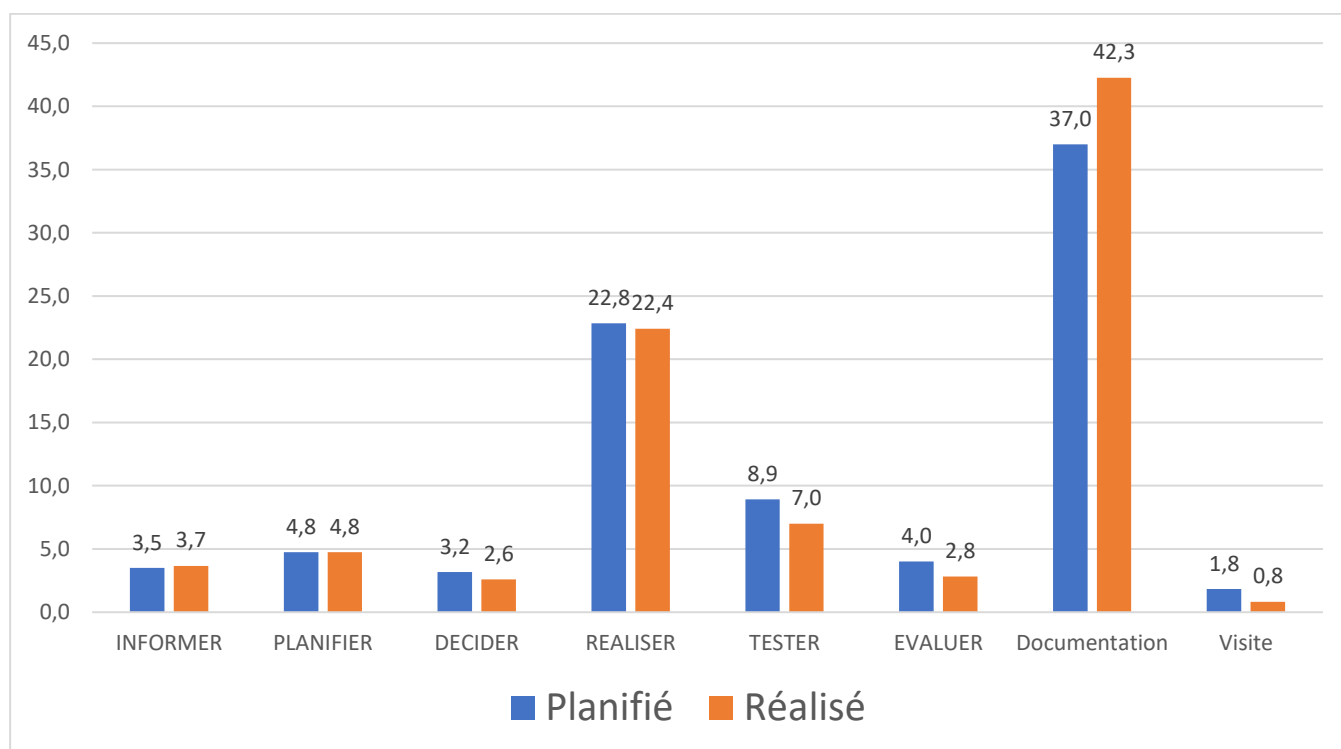
Bilan de planification :

Concernant la planification, nous pouvons voir sur le graphique ci-dessous les principales différences entre la planification et la réalisation. Premièrement, si nous faisons la somme de ces deux valeurs, nous obtenons :

- Planification -> 86h
- Réalisation -> 86h20m

Il y'a uniquement 86 heures de planification car j'ai prévu de finir une demi-journée en avance. La différence de 20 minutes correspond à ce qui s'est réellement passé. J'ai presque une demi-journée d'avance. Cela veut dire que j'ai réalisé le projet en presque exactement le temps planifié.

Concernant le reste des points, nous pouvons remarquer que le point "Documentation" a été pris plus de temps que prévu. Mais avec le temps gagné sur notamment les points "TESTER" et "EVALUER" le tout est compensé.



5 Annexes

5.1 Résumé du rapport du TPI / version succincte de la documentation

Situation de départ :

Le but de ce projet était de proposer un système d'automatisation pour la gestion des comptes AD de l'établissement. Ces actions prennent beaucoup de temps, surtout si cela est fait à la main pour chaque compte utilisateur. C'est pourquoi il est nécessaire d'automatiser ces tâches au maximum afin de limiter l'action humaine et donc aussi limiter les pertes de temps et les erreurs humaines.

Mise en œuvre :

Le programme devait permettre d'automatiser la création, la modification ainsi que la suppression des utilisateurs. Toutes les données devaient être remplies dans des fichiers CSV, puis les scripts après avoir créé ou modifié ou supprimé, générer un nouveau fichier CSV contenant toutes les actions effectuées.

Résultats :

Un script permettant de créer les utilisateurs, un script permettant de modifier les utilisateurs et un script permettant de supprimer ou désactiver les utilisateurs. En plus il y'a 16 scripts utilisées comme des fonctions par les scripts principaux.

5.2 Cahier des charges

ETML Section informatique		Examen - TPI	
1 INFORMATIONS GENERALES			
Candidat :	Nom : PHILIPONA	Prénom : Sylvain	
	✉ : sylvain.philipona@eduvaud.ch	☎ : 077 478 90 42	
Lieu de travail :	<input type="checkbox"/> ETML, Rue de Sébeillon 12, 1004 Lausanne <input type="checkbox"/> CFPV (COFOP-ETML), Avenue de Valmont 28b, 1010 Lausanne		
Orientation :	<input type="checkbox"/> 88601 Développement d'application <input checked="" type="checkbox"/> 88602 Informatique d'entreprise <input type="checkbox"/> 88603 Technique des systèmes		
Chef de projet :	Nom : FERRARI	Prénom :	
	✉ : roberto.ferrari@eduvaud.ch	☎ : 021 316 02 63	
Expert 1 :	Nom : CERAN	Prénom : Suleyman	
	✉ : suleyman.ceran@gmail.com	☎ : 079 396 31 19	
Expert 2 :	Nom : BERNEY	Prénom : Daniel	
	✉ : daniel.berney@heig-vd.ch	☎ : 079 209 87 93	
Période de réalisation :	Du lundi 8 mai 2023 à 8h au mercredi 7 juin 2023 à 16h35		
Horaire de travail :	Lundi	08h00-11h25	13h10-16h35 <i>Pentecôte 29 mai</i>
	Mardi	-	-
	Mercredi	08h00-12h15	13h10-16h35
	Jeudi	-	13h10-16h35 <i>Ascension 18 mai</i>
	Vendredi	08h00-12h15	13h10-16h35 <i>Examen CG 1^{er} juin</i>
	<i>Pont de l'Ascension 19 mai</i> <i>Toutes les demi-journées ont une pause obligatoire de 15 minutes.</i>		
Nombre d'heures :	89 heures		
Planning (en H ou %)	Analyse 10%, Implémentation 50%, Tests 20%, Documentation 20%		
Présentation :	Dates retenues : 14 ou 15 juin 2023		
2 PROCÉDURE			
<p>Le candidat réalise un travail personnel sur la base d'un cahier des charges reçu le 1^{er} jour.</p> <p>Le cahier des charges est approuvé par les deux experts. Il est en outre présenté, commenté et discuté avec le candidat. Par sa signature, le candidat accepte le travail proposé.</p> <p>Le candidat a connaissance de la feuille d'appréciation avant de débiter le travail.</p> <p>Le candidat est entièrement responsable de la sécurité de ses données.</p> <p>En cas de problèmes graves, le candidat avertit au plus vite les deux experts et son CdP.</p> <p>Le candidat a la possibilité d'obtenir de l'aide, mais doit le mentionner dans son dossier.</p> <p>A la fin du délai imparti pour la réalisation du TPI, le candidat doit transmettre par courrier électronique le dossier de projet aux deux experts et au chef de projet. En parallèle, une copie papier du rapport doit être fournie sans délai en trois exemplaires (L'un des deux experts peut demander à ne recevoir que la version électronique du dossier). Cette dernière doit être en tout point identique à la version électronique.</p>			
3 TITRE			
Gestion des utilisateurs dans l'AD de l'établissement.			
Fichier : TPI-RFA-SylvainPhilipona-CdC-V2.3.docx		Version 15 du 24.04.2023 16:21:00	
Auteur :		Dernière modification le 04.05.2023	
		Imprimé le 04.05.2023 14:16:00 à 14:16	

Figure 89 Page 1 CDC



4 MATÉRIEL ET LOGICIEL À DISPOSITION

- 1 ordinateur type ETML avec tous les logiciels standards ETML comme les suites Office et Adobe, PowerShell ISE etc...
- Une machine virtuelle avec un serveur Windows (machine préparée en pré-TPI).
- Et tout autre logiciel nécessaire au bon déroulement du TPI selon les besoins.
- Fichiers .CSV pour la création, la modification et la suppression des comptes.

5 PRÉREQUIS

- Compétences en Scripting → Module 122 (PowerShell).
- Compétences en OS et Serveurs → Modules 305, 123 et 127 (MS Windows 10 et Server 2k19).
- Compétences en outils bureautiques → Module 302.
- Compétences en gestion de projets → Modules 306 et 431

6 DESCRIPTIF DU PROJET

L'ETML, compte plus de 1200 étudiants et apprentis.

Chaque année on compte près de 500 nouveaux arrivants. En début d'année scolaire, il est nécessaire de gérer les comptes utilisateurs des élèves et apprentis. Il faut :

- Créer les comptes des nouveaux élèves et apprentis.
- Mettre à jour les comptes des élèves et des apprentis en cours de formation.
- Supprimer les comptes des élèves et des apprentis qui ont terminé leur cursus.

Afin de pouvoir réaliser ces manipulations, le candidat va créer un ou plusieurs scripts qui, à partir de fichiers CSV, permettent de réaliser les créations, modifications et suppressions de comptes en masse. Un fichier CSV est fourni au candidat en début de TPI.

Pour la création des comptes :

- Le script recherche si l'utilisateur existe déjà dans l'AD. La recherche se fait sur le prénom et le nom de l'utilisateur.
- Le nom d'ouverture de session (identifiant) doit être généré automatiquement à partir des 3 premiers caractères du prénom ainsi que les (maximum) 8 premiers caractères du nom en ayant pris soin en amont de supprimer les caractères diacritiques, les espaces et les apostrophes.
- Le script doit contrôler que le nom d'ouverture de session n'existe pas déjà, en proposer un autre le cas échéant.
- Le mot de passe doit être généré de manière aléatoire et doit être changé par l'utilisateur à la première connexion.
- Le script d'ouverture de session doit être défini automatiquement en fonction de la profession de l'utilisateur.
- Une description doit être générée à partir de la classe de l'utilisateur.
- Le compte doit être créé dans la bonne OU.
- Le dossier personnel (home directory) de l'utilisateur doit être créé au bon endroit et avec les bons droits.
- Le compte doit être membre des différents groupes à partir des TAGs listés dans le fichier CSV.
- Le script doit générer un fichier de sortie CSV contenant toutes les informations nécessaires (identifiant, mot de passe, compte créé ou pas et pourquoi).

Pour la modification des comptes :

- Le script recherche l'utilisateur par son nom d'ouverture de session écrit dans le fichier CSV.



- Une comparaison est faite entre les informations du CSV et celles qui se trouvent dans le compte utilisateur, les différences sont modifiées dans le compte utilisateur.
- Le script doit générer un fichier de sortie CSV contenant les modifications réalisées.

Pour la suppression des comptes :

- Le script recherche l'utilisateur par son nom d'ouverture de session écrit dans le fichier CSV.
- Il doit être possible de choisir si le compte est supprimé ou simplement désactivé.
- Le compte ne peut être supprimé que s'il a été désactivé auparavant.
- Lorsque le compte est supprimé, son dossier personnel (home directory) doit l'être aussi.
- Le script doit générer un fichier de sortie CSV contenant les modifications/suppressions réalisées ou pas.

Méthodologie de tests :

La méthode de tests est laissée au soin du candidat, mais elle doit être reproductible.

7 LIVRABLES

Le candidat est responsable de livrer à son chef de projet et aux deux experts :

- Une planification initiale au format PDF à la fin du premier jour de travail.
- A la fin du TPI, son rapport de projet final et son journal de travail au format PDF.
- A la fin du TPI, une archive contenant les scripts complets (données sources, code source, etc.)

8 POINTS TECHNIQUES ÉVALUÉS SPÉCIFIQUES AU PROJET

La grille d'évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, ...).

En plus de cela, le travail sera évalué sur les 7 points spécifiques suivants (Point A14 à A20):

1. Les scripts sont commentés.
2. Les comptes sont créés correctement, il est possible de se loguer, il est obligatoire de devoir changer son mot de passe.
3. Les dossiers personnels (home directory) sont créés avec les bons droits et sont montés au login du compte.
4. Les comptes sont modifiés correctement.
5. Les comptes sont supprimés / désactivés correctement.
6. Lors de la suppression du compte, le dossier personnel est supprimé correctement.
7. Les fichiers de sortie CSV contiennent toutes les informations nécessaires pour savoir ce qui a été créé, modifié, désactivé, supprimé.

Remarque :

Le recours à des outils en ligne d'intelligence artificielle (ex. : Chat GPT) doit être mentionné et ne peut servir que d'inspiration à la réalisation. En cas d'abus, l'évaluation du TPI en tiendra compte.

5.3 Table d'illustrations

Figure 1 Image de titre	1
Figure 2 Planification initiale	5
Figure 3 Schéma méthode des 6 pas	6
Figure 4 Schéma du processus de création des comptes	12
Figure 5 Schéma du processus de modification des comptes	13
Figure 6 Schéma du processus de suppression des comptes	14
Figure 7 Schéma réseau de l'environnement	15
Figure 8 Format de nommage de variables	16
Figure 9 Format de nommage de paramètres	16
Figure 10 Format de nommage de constantes	16
Figure 11 Format de nommage des scripts	17
Figure 12 Format de nommage des tests unitaires	17
Figure 13 Format d'entête de scripts	18
Figure 14 Format de commentaires	18
Figure 15 Diagramme de flux de la création des comptes	19
Figure 16 Exemple de données dans le CSV	20
Figure 17 Division des groupes avec le caractère "%"	20
Figure 18 Exemple d'exécution du script de création avec des paramètres	21
Figure 19 Vérification existence et format du CSV	21
Figure 20 Vérification des en-têtes du fichier	22
Figure 21 Fonction de vérification des en-têtes du fichier	22
Figure 22 Recherche d'un utilisateur par son nom et prénom	22
Figure 23 Suppression des caractères diacritiques	23
Figure 24 Fonction de suppression des caractères diacritiques	23
Figure 25 Génération de l'identifiant de connexion	24
Figure 26 Régénération si l'identifiant de connexion existe déjà	24
Figure 27 Liste des caractères utilisables	25
Figure 28 Génération aléatoire d'un caractère	25
Figure 29 Exclusion des caractères similaires	26
Figure 30 Génération de la description	26
Figure 31 Dictionnaire des professions	26
Figure 32 Attribution du script	27
Figure 33 Définition des données utilisateur	27
Figure 34 Création et activation du compte	28
Figure 35 L'utilisateur doit changer de mot de passe	28
Figure 36 Création du dossier principal	29
Figure 37 Droits d'accès au dossier	30
Figure 38 Création du répertoire personnel	30
Figure 39 Chemin du dossier personnel	31
Figure 40 Lecteur réseau	31
Figure 41 Exemples de chaîne de caractères de groupes	31
Figure 42 Obtention de la liste des groupes	31
Figure 43 Ajout dans les groupes	31
Figure 44 Fonction d'ajout dans les groupes	31
Figure 45 Déclaration du dictionnaire	32
Figure 46 Erreur utilisateur existe déjà	32
Figure 47 Erreur lors de la création de l'utilisateur	33

Figure 48 Compte créé avec succès	33
Figure 49 Données CSV exportés	33
Figure 50 Diagramme de flux de la modification des comptes	34
Figure 51 Exemple de données dans le CSV	35
Figure 52 Division des groupes avec le caractère "%"	35
Figure 53 Exemple d'exécution du script de modification avec des paramètres	36
Figure 54 Vérification existence et format du CSV	36
Figure 55 Vérification des en-têtes du fichier	37
Figure 56 Fonction de vérification des en-têtes du fichier	37
Figure 57 Recherche d'un utilisateur	37
Figure 58 Génération de la description	38
Figure 59 Dictionnaire des professions	38
Figure 60 Attribution du script	38
Figure 61 Définition des données utilisateur	39
Figure 62 Modification du compte	39
Figure 63 Exemples de chaîne de caractères de groupes	40
Figure 64 Retirer l'utilisateur des groupes	40
Figure 65 Obtention de la liste des groupes	40
Figure 66 Ajout dans les groupes	40
Figure 67 Fonction d'ajout dans les groupes	40
Figure 68 Déclaration du dictionnaire	41
Figure 69 Erreur utilisateur n'existe pas	41
Figure 70 Erreur lors de la modification de l'utilisateur	41
Figure 71 Compte modifié avec succès	42
Figure 72 Données CSV exportés	42
Figure 73 Diagramme de flux de la modification des comptes	43
Figure 74 Exemple de données dans le CSV	44
Figure 75 Exemple d'exécution du script de suppression avec des paramètres	45
Figure 76 Vérification existence et format du CSV	46
Figure 77 Vérification des en-têtes du fichier	46
Figure 78 Fonction de vérification des en-têtes du fichier	46
Figure 79 Recherche d'un utilisateur	47
Figure 80 Désactivation d'un compte utilisateur	47
Figure 81 Vérification si l'utilisateur est désactivé	47
Figure 82 Suppression du répertoire personnel	48
Figure 83 Suppression de l'utilisateur	48
Figure 84 Déclaration du dictionnaire	48
Figure 85 Erreur utilisateur n'existe pas	49
Figure 86 Erreur utilisateur pas désactivé	49
Figure 87 Erreur problèmes avec les champs "Disable" et "Delete"	49
Figure 88 Utilisateur supprimé mais il n'avait pas de répertoire personnel	49
Figure 89 Page 1 CDC	59
Figure 90 Page 2 CDC	60
Figure 91 Page 3 CDC	61

5.4 Sources – Bibliographie

Utilisation	Source
Création de diagrammes	<ul style="list-style-type: none"> • https://lucid.app/lucidchart/63baee3f-0f2c-4eae-bee7-59a599290ba7/edit
Outil d'accessibilité des couleurs	<ul style="list-style-type: none"> • https://www.graphiste-libre.com/outils-daccessibilite-des-couleurs/
Conversions de chaîne de caractères pour les parcourir un par un	<ul style="list-style-type: none"> • https://learn.microsoft.com/en-us/dotnet/api/system.string.tochararray?view=net-7.0
Génération de nombre aléatoire en PowerShell	<ul style="list-style-type: none"> • https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/get-random?view=powershell-7.3
Utilisation du module Pester pour les tests unitaires	<ul style="list-style-type: none"> • https://pester.dev/docs/v4/usage/assertions
Utilisation du module Pester pour les tests unitaires	<ul style="list-style-type: none"> • https://pester.dev/docs/quick-start
Verbes approuvés pour les noms de fonctions PowerShell	<ul style="list-style-type: none"> • https://learn.microsoft.com/en-us/powershell/scripting/developer/cmdlet/approved-verbs-for-windows-powershell-commands?view=powershell-7.3
Catégories Unicode un .NET	<ul style="list-style-type: none"> • https://learn.microsoft.com/en-us/dotnet/api/system.globalization.unicodecategory?view=net-8.0
Formes de normalisation Unicode	<ul style="list-style-type: none"> • https://unicode.org/reports/tr15/
Normalisation de chaînes de caractères	<ul style="list-style-type: none"> • https://learn.microsoft.com/fr-fr/dotnet/api/system.string.normalize?view=net-8.0
Suppressions de diacritiques sur une chaîne de caractères.	<ul style="list-style-type: none"> • https://lazywinadmin.com/2015/05/powershell-remove-diacritics-accents.html
Obtention d'utilisateur AD en PowerShell	<ul style="list-style-type: none"> • https://learn.microsoft.com/en-us/powershell/module/activedirectory/get-aduser?view=windowsserver2022-ps
Vérification si un utilisateur est membre d'un groupe	<ul style="list-style-type: none"> • https://morgantechspace.com/2015/07/powershell-check-if-ad-user-is-member-of-group.html
Conversion de tableau PowerShell en fichier CSV	<ul style="list-style-type: none"> • https://lazyadmin.nl/powershell/get-aduser/https://stackoverflow.com/questions/24311293/powershell-system-array-to-csv-file

Utilisation	Source
Paramètres PowerShell pour accepter plusieurs types	<ul style="list-style-type: none"> • https://www.reddit.com/r/PowerShell/comments/6k9foq/function_parameter_accept_username_string_or/
Différence entre les propriétés AD "samAccountName" et "userPrincipalName"	<ul style="list-style-type: none"> • https://www.it-connect.fr/active-directory-samaccountname-vs-userprincipalname/
Match de valeurs dans un dictionnaire	<ul style="list-style-type: none"> • https://stackoverflow.com/questions/39906809/powershell-2-0-how-to-match-or-like-a-string-in-hash-table-keys
Problèmes avec le paramètre forçant l'utilisateur à changer de mot de passe	<ul style="list-style-type: none"> • https://stackoverflow.com/questions/35141099/changepassword-atlogin-not-applying-on-new-aduser-when-enabled-is-false
Création d'utilisateur AD en PowerShell	<ul style="list-style-type: none"> • https://learn.microsoft.com/en-us/powershell/module/activedirectory/new-aduser?view=windowsserver2022-ps
Vérifier l'extension d'un fichier en PowerShell	<ul style="list-style-type: none"> • https://stackoverflow.com/questions/28787364/check-file-extension
Conventions de nommage de scripts PowerShell	<ul style="list-style-type: none"> • https://www.interfacett.com/blogs/how-should-i-name-my-powershell-scripts/
Diagrammes de flux	<ul style="list-style-type: none"> • https://miro.com/app/board/uXjVMFDguGw=/
Formater une chaîne de caractères	<ul style="list-style-type: none"> • https://stackoverflow.com/questions/36585500/how-to-display-formatted-output-with-write-host
Retirer un utilisateur AD de tous ses groupes	<ul style="list-style-type: none"> • https://stackoverflow.com/questions/44144678/trying-to-remove-user-from-all-groups-in-an-active-directory-using-powershell-sc
Création de diagrammes type Visio	<ul style="list-style-type: none"> • https://app.diagrams.net/
Trier une table Word alphabétiquement	<ul style="list-style-type: none"> • https://support.microsoft.com/en-gb/office/sort-a-list-alphabetically-in-word-4d27ca57-6d64-4229-82f8-a0a1a805d494
Image de titre	<ul style="list-style-type: none"> • https://turnoff.us/

5.5 Journal de travail

Séquence 1			Date: lundi, 8 mai 2023	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
INFORMER : Analyse du projet	17	1h25min	<ul style="list-style-type: none"> Lecture et validation du cahier des charges avec M.Ceran. Création du répertoire GitHub 	
PLANIFIER : Planification initiale	21	1h45min	<ul style="list-style-type: none"> Planification initiale 	
Total tranche	38	3h10min		
Séquence 2			Date: lundi, 8 mai 2023	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
PLANIFIER : Planification initiale	36	3h	<ul style="list-style-type: none"> Planification initiale 	
INFORMER : Analyse du projet	2	0h10min	<ul style="list-style-type: none"> Discussion avec M. Ferrari au sujet du cahier des charges 	
Total tranche	38	3h10min		
Séquence 3			Date: mercredi, 10 mai 2023	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Documentation - Journal de travail	1	0h5min	<ul style="list-style-type: none"> Remplissage du journal de travail 	
Documentation - Rapport	22	1h50min	<ul style="list-style-type: none"> Introduction du projet Objectifs du projet Planification initiale (en cours) Conception <p>La conception me donne une vision plus globale du projet. Cela me rassure dans la faisabilité du projet</p>	<ul style="list-style-type: none"> https://www.graphiste-libre.com/outils-daccessibilite-des-couleurs/
INFORMER : Analyse du projet	25	2h5min	<ul style="list-style-type: none"> Analyse et conception de la méthode pour la création des comptes Analyse et conception de la méthode pour la modification des comptes Analyse et conception de la méthode pour la suppression des comptes 	<ul style="list-style-type: none"> https://lucid.app/lucidchart/63baee3f-0f2c-4eae-bee7-59a599290ba7/edit?viewport_loc=-444%2C-136%2C4039%2C2017%2CeQDWfm~.QisA&invitationId=in
Total tranche	48	4h		

Séquence 4			Date: mercredi, 10 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min	• Remplissage du journal de travail		
Documentation - Rapport	6	0h30min	• Conception (en cours)		
DECIDER : Choix de la solution	1	0h5min	• Création d'un script .PS1 prenant en input 2 arguments (Prénom + Nom) et retournant l'identifiant de login		
REALISER - Création : Génération identifiant de login	6	0h30min	• Création un script .PS1 prenant en input 2 arguments (Prénom + Nom) et retournant l'identifiant de login sous le format 3 premiers caractere du prénom + les 8 premiers caractere du nom La tâche à pris moins de temps que prévu. Cela est rassurant pour la suite du projet	<ul style="list-style-type: none"> • https://learn.microsoft.com/en-us/powershell/scripting/developer/cmdlet/approved-verbs-for-windows-powershell-commands?view=powershell-7.3 	
TESTER : Test de la solution	4	0h20min	• Création et execution de tests unitaires pour tester multiples cas d'inputs	<ul style="list-style-type: none"> • https://pester.dev/docs/quick-start • https://pester.dev/docs/v4/usage/assertions 	
EVALUER : Validation de la solution selon le CDC	2	0h10min	• La solution correspond au CDC dans le cas où un autre script va être implémenté par la suite qui va supprimer les caractères diacritiques et les espaces et apostrophes		
DECIDER : Choix de la solution	1	0h5min	• Création d'un script .PS1 prenant en argument un nombre et retournant un mot de passe aléatoire de la longueur de la valeur de cet argument		
REALISER - Création : Génération mot de passe	9	0h45min	• Création d'un script .PS1 prenant en argument un nombre et retournant un mot de passe aléatoire de la longueur de la valeur de cet argument • Un paramètre 'ExcludeSimilar' permet d'éviter d'avoir des caractères qui pourraient être confondus avec d'autres	<ul style="list-style-type: none"> • https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/get-random?view=powershell-7.3 	
TESTER : Test de la solution	5	0h25min	• Création et execution de tests unitaires pour tester multiples cas d'inputs	<ul style="list-style-type: none"> • https://learn.microsoft.com/en-us/dotnet/api/system.string.tochararray?view=net-7.0 	
EVALUER : Validation de la solution selon le CDC	2	0h10min	• La solution correspond au CDC		
Total tranche	38	3h10min			

Séquence 5			Date: jeudi, 11 mai 2023	Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Documentation - Journal de travail	2	0h10min		
Documentation - Rapport	12	1h	• Conception (en cours)	
DECIDER : Choix de la solution	1	0h5min	• Création d'un script .PS1 prenant en input 2 arguments (Prénom + Nom) et retournant si l'utilisateur existe.	
REALISER : Création : Recherche utilisateur existant	3	0h15min	• Création d'un script .PS1 prenant en input 2 arguments (Prénom + Nom) et retournant si l'utilisateur existe. Le filtre se fait sur les champs ad "GivenName" et "sn"	
TESTER : Test de la solution	3	0h15min	• Test du script sur le serveur AD avec plusieurs cas de Nom + Prénom. • Les tests sont concluants	
EVALUER : Validation de la solution selon le CDC	1	0h5min	• La solution correspond au CDC	
REALISER - Création : Génération mot de passe	1	0h5min	• Optimisation de code lors de l'exclusion des caractères similaires	
TESTER : Test de la solution	1	0h5min	• Execution des tests unitaires pour vérifier que le script fonctionne correctement • Les tests sont concluants	• https://learn.microsoft.com/en-us/powershell/module/activedirectory/get-aduser?view=windowsserver2022-ps
DECIDER : Choix de la solution	3	0h15min	• Création d'un script .PS1 prenant en input une chaîne de caractères et retourne la chaîne en ayant remplacé les caractères diacritiques par des caractères sans accents, cédilles...	
REALISER - Création : Suppression des diacritiques	7	0h35min	• Création d'un script .PS1 prenant en input une chaîne de caractères et retourne la chaîne en ayant remplacé les caractères diacritiques par des caractères sans accents, cédilles • Le script utilise la normalisation unicode pour séparer les caractères accentués, ect sous la forme suivante : é => e ' • Exemple : L'été de Raphaël => L ' e ' t e ' d e R a p h a e l • Tout ces caractères spéciaux sont ensuite enlevés. Ainsi que les espaces et les apostrophes	• https://lazywinadmin.com/2015/05/powershell-remove-diacritics-accents.html • https://learn.microsoft.com/fr-fr/dotnet/api/system.string.normalize?view=net-8.0 • https://unicode.org/reports/tr15/ • https://learn.microsoft.com/en-us/dotnet/api/system.globalization.unicodetextcategory?view=net-8.0
TESTER : Test de la solution	3	0h15min	• Création et execution de tests unitaires pour tester multiples cas d'inputs • Les tests sont concluants	
EVALUER : Validation de la solution selon le CDC	1	0h5min	• La solution correspond au CDC	

Séquence 6			Date: vendredi, 12 mai 2023	Matin - 4H00
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Documentation - Journal de travail	2	0h10min		
Visite - Chef de projet / Experts	2	0h10min	• Visite hebdomadaire de Monsieur Ferrari	
DECIDER : Choix de la solution	1	0h5min	• Création d'un script .PS1 prenant en input la classe de l'utilisateur et retourne une description basée sur une variable dans le script	
REALISER - Création : Génération description	2	0h10min	• Création d'un script .PS1 prenant en input la classe de l'utilisateur et retourne une description basée sur une variable dans le script.	
TESTER : Test de la solution	1	0h5min	• Execution des tests unitaires pour vérifier que le script fonctionne correctement • Les tests sont concluents	
EVALUER : Validation de la solution selon le CDC	1	0h5min	• La solution correspond au CDC	
DECIDER : Choix de la solution	3	0h15min	• Création d'un script .PS1 prenant en input un utilisateur AD et une liste de groupes	
REALISER - Création : Ajout dans les groupes	24	2h	• Création d'un script .PS1 prenant en input un utilisateur AD et une liste de groupes • Le script parcourt tout les groupes pour ajouter l'utilisateur dedans • Un tableau est retourné avec les actions effectuées pour chaque groupe	<ul style="list-style-type: none"> • https://stackoverflow.com/questions/39906809/powershell-2-0-how-to-match-or-like-a-string-in-hash-table-keys • https://www.reddit.com/r/PowerShell/comments/6k9foq/function_parameter_accept_username_string_or/ • https://lazyadmin.nl/powershell/get-aduser/https://stackoverflow.com/questions/24311293/powershell-system-array-to-csv-file • https://morgantechspace.com/2015/07/powershell-check-if-ad-user-is-member-of-group.html
TESTER : Test de la solution	4	0h20min	• Test du script sur le serveur AD avec plusieurs cas d'utilisateur AD + groupes. • Les tests sont concluents	
EVALUER : Validation de la solution selon le CDC	2	0h10min	• La solution correspond au CDC	
DECIDER : Choix de la solution	1	0h5min	• Création d'un script .ps1 prenant en input le nom de la profession de l'utilisateur, la compare avec une table de vérité et retourne le nom du script	
REALISER - Création : Script d'ouverture de session	5	0h25min	• Création d'un script .ps1 prenant en input le nom de la profession de l'utilisateur, la compare avec une table de vérité et retourne le nom du script	
Total tranche	48	4h		

Séquence 7			Date: vendredi, 12 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Rapport	20	1h40min			
REALISER - Création : Script d'ouverture de session	7	0h35min		<ul style="list-style-type: none"> • https://stackoverflow.com/questions/39906809/powershell-2-0-how-to-match-or-like-a-string-in-hash-table-keys 	
TESTER : Test de la solution	2	0h10min	<ul style="list-style-type: none"> • Execution des tests unitaires pour vérifier que le script fonctionne correctement • Les tests sont concluents 		
EVALUER : Validation de la solution selon le CDC	1	0h5min	<ul style="list-style-type: none"> • La solution correspond au CDC 		
DECIDER : Choix de la solution	1	0h5min	<ul style="list-style-type: none"> • Création d'un script .PS1 prenant en input un login et retournant si le login existe. 		
REALISER - Création : Verification existence identifiant de login	2	0h10min	<ul style="list-style-type: none"> • Création d'un script .PS1 prenant en input un login et retournant si le login existe. • Le filtre se fait sur le champ ad "samAccountName" 	<ul style="list-style-type: none"> • https://www.it-connect.fr/active-directory-samaccountname-vs-userprincipalname/ 	
TESTER : Test de la solution	2	0h10min	<ul style="list-style-type: none"> • Test du script sur le serveur AD avec plusieurs cas d'utilisateur AD. • Les tests sont concluents 		
EVALUER : Validation de la solution selon le CDC	1	0h5min	<ul style="list-style-type: none"> • La solution correspond au CDC 		
Total tranche	38	3h10min			

Séquence 8			Date: lundi, 15 mai 2023		Matin - 3H10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Rapport	19	1h35min	<ul style="list-style-type: none"> Stratégie de tests Dossier de conception 		
DECIDER : Choix de la solution	2	0h10min	<ul style="list-style-type: none"> Création d'un script .PS1 prenant en input le chemin du dossier principal des homes directory, le nom du partage et le nom de l'utilisateur à qui créer le dossier home 		
REALISER - Création : Création HomeDirectory	15	1h15min	<ul style="list-style-type: none"> Création d'un script .PS1 prenant en input le chemin du dossier principal des homes directory, le nom du partage et le nom de l'utilisateur à qui créer le dossier home Le script vérifie si le dossier principal des homes existe, et dans le cas ou il n'existe pas, le crée, le partage et lui affecte les bon droits 		
Total tranche	38	3h10min			

Séquence 9			Date: lundi, 15 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Rapport	36	3h	<ul style="list-style-type: none"> Stratégie de tests Dossier de conception 		
Total tranche	38	3h10min			
Séquence 10			Date: mercredi, 17 mai 2023		Matin - 4H00
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Rapport	13	1h5min			
REALISER - Création : Création HomeDirectory	25	2h5min	<ul style="list-style-type: none"> Création d'un script .PS1 prenant en input le chemin du dossier principal des homes directory, le nom du partage et le nom de l'utilisateur à qui créer le dossier home Le script vérifie si le dossier principal des homes existe, et dans le cas ou il n'existe pas, le crée, le partage et lui affecte les bon droits 		
TESTER : Test de la solution	6	0h30min	<ul style="list-style-type: none"> Test du script sur le serveur AD avec plusieurs cas d'utilisateurs Les tests sont concluents 		
EVALUER : Validation de la solution selon le CDC	2	0h10min	<ul style="list-style-type: none"> La solution correspond au CDC 		

Séquence 11			Date: mercredi, 17 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
DECIDER : Choix de la solution	3	0h15min	• Création d'un script principal .ps1 qui va lancer tous les autres scripts pour effectuer la création des comptes selon le CDC		
REALISER - Création : Intégration	33	2h45min	• Intégration de tous les scripts entre eux • Cette tâche à pris plus de temps que prévu mais je reste quand même dans les temps		
Total tranche	38	3h10min			
Séquence 12			Date: lundi, 22 mai 2023		Matin - 3H10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
REALISER - Création : Intégration	29	2h25min	• Importation selon le délimiteur CSV • Régénération du login si celui-ci existe déjà • Demande à l'utilisateur de se changer son mot de passe lors de la première connexion • Crée l'utilisateur dans la bonne OU	<ul style="list-style-type: none"> • https://stackoverflow.com/questions/28787364/check-file-extension • https://learn.microsoft.com/en-us/powershell/module/activedirectory/new-aduser?view=windowsserver2022-ps • https://stackoverflow.com/questions/35141099/change-passwordatlogin-not-applying-on-new-aduser-when-enabled-is-false 	
TESTER : Test de la solution	7	0h35min	• Test du script sur le serveur AD avec plusieurs cas de données • Les tests sont concluents		
Total tranche	38	3h10min			

Séquence 13			Date: lundi, 22 mai 2023	Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Documentation - Journal de travail	2	0h10min		
Visite - Chef de projet / Experts	4	0h20min	• Visite de M. Berney	
EVALUER : Validation de la solution selon le CDC	2	0h10min	• La solution correspond au CDC	
DECIDER : Choix de la solution	1	0h5min	• Création d'un tableau avec les données suivantes : User, Action, Login, Password, LoginScript, HomeDirectory, Comments, Groups. Toutes les actions effectuées dans le script de création des comptes sont loggés dans ce tableau. • Le tableau est généré pour chaque utilisateur, puis exporté en .CSV	
REALISER - Création : Génération du CSV	24	2h	• Création d'un tableau avec les données suivantes : User, Action, Login, Password, LoginScript, HomeDirectory, Comments, Groups. Toutes les actions effectuées dans le script de création des comptes sont loggés dans ce tableau. • Le tableau est généré pour chaque utilisateur, puis exporté en .CSV	
TESTER : Test de la solution	3	0h15min	• Test du script sur avec plusieurs cas de données • Les tests sont concluents	
EVALUER : Validation de la solution selon le CDC	2	0h10min	• La solution correspond au CDC	
Total tranche	38	3h10min		

Séquence 14			Date: mercredi, 24 mai 2023	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Documentation - Journal de travail	2	0h10min		
Documentation - Rapport	46	3h50min	<ul style="list-style-type: none"> Dossier de réalisation : Création des comptes Dossier de réalisation : Environnement, version du projet et normes de codage / nommage 	<ul style="list-style-type: none"> https://www.interfacett.com/blogs/how-should-i-name-my-powershell-scripts/
Total tranche	48	4h		
Séquence 15			Date: mercredi, 24 mai 2023	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Documentation - Journal de travail	1	0h5min		
Documentation - Rapport	37	3h5min	<ul style="list-style-type: none"> Dossier de réalisation : Création des comptes Création d'un diagramme de flux de la création des comptes 	<ul style="list-style-type: none"> https://miro.com/app/board/uxjVMFDguGw=/
Total tranche	38	3h10min		

Séquence 16			Date: jeudi, 25 mai 2023	Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Documentation - Journal de travail	2	0h10min		
Documentation - Rapport	9	0h45min	• Dossier de réalisation	
DECIDER : Choix de la solution	1	0h5min	• Pour ce point pas de script nécessaire car il existe déjà une fonction AD permettant de réaliser ce point	
REALISER - Suppression : Recherche de l'utilisateur	1	0h5min	• Création de la commande Get-AdUser filtrant par la champ samAccountName	
TESTER : Test de la solution	1	0h5min	• Test de la commande avec plusieurs cas de données • Les tests sont concluants	
EVALUER : Validation de la solution selon le CDC	2	0h10min	• La solution correspond au CDC	
DECIDER : Choix de la solution	1	0h5min	• Création d'un script .PS1 prenant en argument l'identifiant de login de l'utilisateur	
REALISER - Suppression : Suppression de l'utilisateur	2	0h10min	• Le script effectue une recherche si l'utilisateur existe. Dans ce cas le compte est supprimé et la valeur "True" est retournée. Dans le cas contraire rien n'est fait et la valeur "False" est retournée.	
TESTER : Test de la solution	1	0h5min	• Test du script avec plusieurs cas de données • Les tests sont concluants	
EVALUER : Validation de la solution selon le CDC	1	0h5min	• La solution correspond au CDC	

DECIDER : Choix de la solution	1	0h5min	• Création d'un script .PS1 prenant en argument l'identifiant de login de l'utilisateur	
REALISER - Suppression : Desactivation de l'utilisateur	2	0h10min	• Le script effectue une recherche si l'utilisateur existe. Dans ce cas le compte est désactivé et la valeur "True" est retournée. Dans le cas contraire rien n'est fait et la valeur "False" est retournée.	
TESTER : Test de la solution	1	0h5min	• Test du script avec plusieurs cas de données • Les tests sont concluents	
EVALUER : Validation de la solution selon le CDC	1	0h5min	• La solution correspond au CDC	
DECIDER : Choix de la solution	2	0h10min	• Création d'un script .PS1 prenant en argument l'identifiant de login de l'utilisateur	
REALISER - Suppression : Suppression du HomeDirectory	6	0h30min	• L'utilisateur est recherché par son identifiant de login • Le champ HomeDirectory est récupéré du compte • Le dossier est ensuite supprimé	
TESTER : Test de la solution	3	0h15min	• Test du script avec plusieurs cas de données • Les tests sont concluents	
EVALUER : Validation de la solution selon le CDC	1	0h5min	• La solution correspond au CDC	
Total tranche	38	3h10min		

Séquence 17			Date: vendredi, 26 mai 2023		Matin - 4H00
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
DECIDER : Choix de la solution	1	0h5min	<ul style="list-style-type: none"> Intégration de tous les scripts entre eux 		
REALISER - Suppression : Intégration	12	1h	<ul style="list-style-type: none"> Importation selon le délimiteur CSV Teste si l'utilisateur doit être supprimé ou désactivé Désactive l'utilisateur Supprime l'utilisateur si celui-ci est déjà désactivé 	<ul style="list-style-type: none"> https://stackoverflow.com/questions/36585500/how-to-display-formatted-output-with-write-host 	
TESTER : Test de la solution	7	0h35min	<ul style="list-style-type: none"> Test du script avec plusieurs cas de données Les tests sont concluents 		
EVALUER : Validation de la solution selon le CDC	2	0h10min	<ul style="list-style-type: none"> La solution correspond au CDC 		
DECIDER : Choix de la solution	2	0h10min	<ul style="list-style-type: none"> Création d'un tableau avec les données suivantes : User, Action, Login, Password, LoginScript, HomeDirectory, Comments, Groups. Toutes les actions effectuées dans le script de création des comptes sont loggés dans ce tableau. Le tableau est généré pour chaque utilisateur, puis exporté en .CSV 		
REALISER - Suppression : Génération du CSV	17	1h25min	<ul style="list-style-type: none"> Création d'un tableau avec les données suivantes : User, Action, Login, Password, LoginScript, HomeDirectory, Comments, Groups. Toutes les actions effectuées dans le script de création des comptes sont loggés dans ce tableau. Le tableau est généré pour chaque utilisateur, puis exporté en .CSV 		
TESTER : Test de la solution	3	0h15min	<ul style="list-style-type: none"> Test du script avec plusieurs cas de données Les tests sont concluents 		
EVALUER : Validation de la solution selon le CDC	2	0h10min	<ul style="list-style-type: none"> La solution correspond au CDC 		
Total tranche	48	4h			

Séquence 18			Date: vendredi, 26 mai 2023	Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Documentation - Journal de travail	2	0h10min		
Visite - Chef de projet / Experts	4	0h20min	<ul style="list-style-type: none"> • Visite de M. Ferrari • Réponse à mes questions concernant les format csv : A mon libre choix • Réponse à mes questions concernant la création du home directory : Il faut mettre les droits en Modify aux utilisateurs et pas en Control Total 	
DECIDER : Choix de la solution	1	0h5min	<ul style="list-style-type: none"> • Refactorisation de la génération du csv sous le même format que pour la suppression des comptes 	
REALISER - Création : Génération du CSV	18	1h30min	<ul style="list-style-type: none"> • Gestion des erreurs plus performante et gère des cas supplémentaires 	
TESTER : Test de la solution	7	0h35min	<ul style="list-style-type: none"> • Test du script avec plusieurs cas de données • Les tests sont concluents 	
EVALUER : Validation de la solution selon le CDC	1	0h5min	<ul style="list-style-type: none"> • La solution correspond au CDC 	
DECIDER : Choix de la solution	1	0h5min	<ul style="list-style-type: none"> • Pour ce point pas de script nécessaire car il existe déjà une fonction AD permettant de réaliser ce point 	
REALISER - Suppression : Recherche de l'utilisateur	1	0h5min	<ul style="list-style-type: none"> • Création de la commande Get-AdUser filtrant par la champ samAccountName 	
TESTER : Test de la solution	1	0h5min	<ul style="list-style-type: none"> • Test de la commande avec plusieurs cas de données • Les tests sont concluents 	
EVALUER : Validation de la solution selon le CDC	2	0h10min	<ul style="list-style-type: none"> • La solution correspond au CDC 	
Total tranche	38	3h10min		

Séquence 19			Date: mercredi, 31 mai 2023	Matin - 4H00
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Documentation - Journal de travail	3	0h15min		
REALISER - Création : Intégration	7	0h35min	<ul style="list-style-type: none"> • Ajout du champ E-mail lors de la création des comptes • Changement de droits en "Modify" lors de la création du home directory 	
TESTER : Test de la solution	2	0h10min	<ul style="list-style-type: none"> • Test du script avec plusieurs cas de données • Les tests sont concluents 	
EVALUER : Validation de la solution selon le CDC	1	0h5min	<ul style="list-style-type: none"> • La solution correspond au CDC 	
DECIDER : Choix de la solution	1	0h5min	<ul style="list-style-type: none"> • Création d'un script .PS1 prenant en argument les données à modifier et modifie ensuite l'utilisateur 	
REALISER - Modification : Modifications de l'utilisateur	3	0h15min	<ul style="list-style-type: none"> • Le script récupère les nouvelles données de l'utilisateur • La description et le script de login sont régénérés • Le compte est modifié avec les nouvelles informations fournies dans le CSV 	<ul style="list-style-type: none"> • https://stackoverflow.com/questions/44144678/trying-to-remove-user-from-all-groups-in-an-active-directory-using-powershell-sc
TESTER : Test de la solution	4	0h20min	<ul style="list-style-type: none"> • Test du script avec plusieurs cas de données • Les tests sont concluents 	
EVALUER : Validation de la solution selon le CDC	1	0h5min	<ul style="list-style-type: none"> • La solution correspond au CDC 	

DECIDER : Choix de la solution	1	0h5min	<ul style="list-style-type: none"> • Intégration des scripts et commandes entre eux 	
REALISER - Modification : Intégration	4	0h20min	<ul style="list-style-type: none"> • Importation selon le délimiteur CSV • Teste si l'utilisateur existe • Modifie l'utilisateur 	
TESTER : Test de la solution	6	0h30min	<ul style="list-style-type: none"> • Test du script avec plusieurs cas de données • Les tests sont concluents 	
EVALUER : Validation de la solution selon le CDC	2	0h10min	<ul style="list-style-type: none"> • La solution correspond au CDC 	
DECIDER : Choix de la solution	1	0h5min	<ul style="list-style-type: none"> • Création d'un tableau avec les données suivantes : User, Action, Email, Classe, LoginScript, Comments, Groups. Toutes les actions effectuées dans le script de création des comptes sont loggés dans ce tableau. • Le tableau est généré pour chaque utilisateur, puis exporté en .CSV 	
REALISER - Modification : Génération du CSV	4	0h20min	<ul style="list-style-type: none"> • Création d'un tableau avec les données suivantes : User, Action, Email, Classe, LoginScript, Comments, Groups. Toutes les actions effectuées dans le script de création des comptes sont loggés dans ce tableau. • Le tableau est généré pour chaque utilisateur, puis exporté en .CSV 	
TESTER : Test de la solution	7	0h35min	<ul style="list-style-type: none"> • Test du script avec plusieurs cas de données • Les tests sont concluents 	
EVALUER : Validation de la solution selon le CDC	1	0h5min	<ul style="list-style-type: none"> • La solution correspond au CDC 	
Total tranche	48	4h		

Séquence 20			Date: mercredi, 31 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	1	0h5min			
Documentation - Rapport	37	3h5min	• Dossier de réalisation		
Total tranche	38	3h10min			
Séquence 21			Date: vendredi, 2 juin 2023		Matin - 4H00
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	1	0h5min			
Documentation - Rapport	47	3h55min	• Dossier de réalisation		
Total tranche	48	4h			
Séquence 22			Date: vendredi, 2 juin 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	1	0h5min			
Documentation - Rapport	37	3h5min	• Dossier de réalisation • Création de diagrammes de flux pour la modification et la suppression des comptes	• https://miro.com/app/board/uXjVMFDguGw=/	
Total tranche	38	3h10min			
Séquence 23			Date: lundi, 5 juin 2023		Matin - 3H10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	1	0h5min			
Documentation - Rapport	37	3h5min	• Dossier de réalisation • Schéma d'environnement serveur • Description des tests	• https://app.diagrams.net/	
Total tranche	38	3h10min			

Séquence 24			Date: lundi, 5 juin 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	1	0h5min			
Documentation - Rapport	37	3h5min	<ul style="list-style-type: none"> Description des tests Glossaire Correction de fautes Mise en page 		
Total tranche	38	3h10min			
Séquence 25			Date: mercredi, 7 juin 2023		Matin - 4H00
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	1	0h5min			
Documentation - Rapport	47	3h55min	<ul style="list-style-type: none"> Méthodologie de projet Description de la planification Conclusion Correction de fautes Mise en page 	<ul style="list-style-type: none"> https://support.microsoft.com/en-gb/office/sort-a-list-alphabetically-in-word-4d27ca57-6d64-4229-82f8-a0a1a805d494 	
Total tranche	48	4h			
Séquence 26			Date: mercredi, 7 juin 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Impression et rendu du TPI	6	0h30min	<ul style="list-style-type: none"> Envoi de la version finale du TPI Impression du document 		

5.6 Planification détaillée

Séquence 1			Date: lundi, 8 mai 2023		Matin - 3H10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
INFORMER : Analyse du projet	17	1h25min	<ul style="list-style-type: none"> Lecture et validation du cahier des charges avec M.Ceran. Création du répertoire GitHub Planification initiale 		
PLANIFIER : Planification initiale	21	1h45min			
Total tranche	38	3h10min			
Séquence 2			Date: lundi, 8 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
PLANIFIER : Planification initiale	36	3h	<ul style="list-style-type: none"> Planification initiale 		
INFORMER : Analyse du projet	2	0h10min	<ul style="list-style-type: none"> Discussion avec M.Ferrari au sujet du cahier des charges 		
Total tranche	38	3h10min			
Séquence 3			Date: mercredi, 10 mai 2023		Matin - 4H00
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min	<ul style="list-style-type: none"> Analyse préliminaire Conception Stratégie de tests Dossier de conception 		
Documentation - Rapport	23	1h55min			
INFORMER : Analyse du projet	23	1h55min			
Total tranche	48	4h			

Séquence 4			Date: mercredi, 10 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Rapport	10	0h50min	• Dossier de conception		
DECIDER : Choix de la solution	1	0h5min	• Choix de la solution pour la génération des identifiants de login		
REALISER - Création : Génération identifiant de login	18	1h30min	• Réalisation de la génération des identifiants de login		
TESTER : Test de la solution	4	0h20min	• Tests unitaires de la génération des identifiants de login		
EVALUER : Validation de la solution selon le CDC	3	0h15min	• Validation de la génération des identifiants de login selon les points demandés dans le CDC		
Total tranche	38	3h10min			
Séquence 5			Date: jeudi, 11 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
DECIDER : Choix de la solution	2	0h10min			
REALISER - Création : Génération mot de passe	9	0h45min	• Réalisation de la génération du mot de passe		
TESTER : Test de la solution	4	0h20min	• Réalisation de la génération du mot de passe		
EVALUER : Validation de la solution selon le CDC	2	0h10min	• Tests unitaires de la génération du mot de passe		
DECIDER : Choix de la solution	1	0h5min			
REALISER - Création : Recherche utilisateur existant	9	0h45min			
TESTER : Test de la solution	7	0h35min			
EVALUER : Validation de la solution selon le CDC	2	0h10min			
Total tranche	38	3h10min			

Séquence 6			Date: vendredi, 12 mai 2023	Matin - 4H00
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Documentation - Journal de travail	2	0h10min		
Documentation - Rapport	10	0h50min		
Visite - Chef de projet / Experts	2	0h10min	• Visite chef de projet	
DECIDER : Choix de la solution	1	0h5min		
REALISER - Création : Suppression des diacritiques	18	1h30min		
TESTER : Test de la solution	4	0h20min		
EVALUER : Validation de la solution selon le CDC	1	0h5min		
DECIDER : Choix de la solution	1	0h5min		
REALISER - Création : Verification existence identifiant de login	3	0h15min		
TESTER : Test de la solution	5	0h25min		
EVALUER : Validation de la solution selon le CDC	1	0h5min		
Total tranche	48	4h		

Séquence 7			Date: vendredi, 12 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
DECIDER : Choix de la solution	1	0h5min			
REALISER - Création : Génération description	3	0h15min			
TESTER : Test de la solution	3	0h15min			
EVALUER : Validation de la solution selon le CDC	1	0h5min			
DECIDER : Choix de la solution	1	0h5min			
REALISER - Création : Ajout dans les groupes	19	1h35min			
TESTER : Test de la solution	6	0h30min			
EVALUER : Validation de la solution selon le CDC	2	0h10min			
Total tranche	38	3h10min			
Séquence 8			Date: lundi, 15 mai 2023		Matin - 3H10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Rapport	36	3h			

Séquence 9			Date: lundi, 15 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
DECIDER : Choix de la solution	2	0h10min			
REALISER - Création : Création HomeDirectory	14	1h10min			
TESTER : Test de la solution	4	0h20min			
EVALUER : Validation de la solution selon le CDC	2	0h10min			
DECIDER : Choix de la solution	2	0h10min			
REALISER - Création : Script d'ouverture de session	12	1h			
Total tranche	38	3h10min			
Séquence 10			Date: mercredi, 17 mai 2023		Matin - 4H00
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Rapport	12	1h			
REALISER - Création : Script d'ouverture de session	12	1h			
TESTER : Test de la solution	8	0h40min			
EVALUER : Validation de la solution selon le CDC	4	0h20min			
DECIDER : Choix de la solution	1	0h5min			
REALISER - Création : Intégration	9	0h45min			
Total tranche	48	4h			

Séquence 11			Date: mercredi, 17 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Rapport	22	1h50min			
Visite - Chef de projet / Experts	2	0h10min	• Visite chef de projet		
REALISER - Création : Intégration	12	1h			
Total tranche	38	3h10min			
Séquence 12			Date: lundi, 22 mai 2023		Matin - 3H10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
TESTER : Test de la solution	12	1h			
EVALUER : Validation de la solution selon le CDC	4	0h20min			
DECIDER : Choix de la solution	1	0h5min			
REALISER - Création : Génération du CSV	14	1h10min			
TESTER : Test de la solution	3	0h15min			
EVALUER : Validation de la solution selon le CDC	2	0h10min			

Séquence 13			Date: lundi, 22 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Rapport	34	2h50min			
Visite - Chef de projet / Experts	2	0h10min	• Visite chef de projet		
Total tranche	38	3h10min			
Séquence 14			Date: mercredi, 24 mai 2023		Matin - 4H00
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
DECIDER : Choix de la solution	2	0h10min			
REALISER - Modification : Recherche de l'utilisateur	8	0h40min			
TESTER : Test de la solution	4	0h20min			
EVALUER : Validation de la solution selon le CDC	2	0h10min			
DECIDER : Choix de la solution	2	0h10min			
REALISER - Modification : Modifications de l'utilisateur	18	1h30min			
TESTER : Test de la solution	6	0h30min			
TESTER : Test de la solution	2	0h10min			
EVALUER : Validation de la solution selon le CDC	2	0h10min			
Total tranche	48	4h			

Séquence 15			Date: mercredi, 24 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
DECIDER : Choix de la solution	3	0h15min			
REALISER - Modification : Intégration	22	1h50min			
TESTER : Test de la solution	8	0h40min			
EVALUER : Validation de la solution selon le CDC	3	0h15min			
Total tranche	38	3h10min			
Séquence 16			Date: jeudi, 25 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Rapport	27	2h15min			
DECIDER : Choix de la solution	1	0h5min			
REALISER - Modification : Génération du CSV	4	0h20min			
TESTER : Test de la solution	2	0h10min			
EVALUER : Validation de la solution selon le CDC	2	0h10min			
Total tranche	38	3h10min			

Séquence 17			Date: vendredi, 26 mai 2023		Matin - 4H00
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
DECIDER : Choix de la solution	2	0h10min			
REALISER - Suppression : Recherche de l'utilisateur	8	0h40min			
TESTER : Test de la solution	4	0h20min			
EVALUER : Validation de la solution selon le CDC	2	0h10min			
DECIDER : Choix de la solution	4	0h20min	• Choix du format du CSV pour la suppression		
REALISER - Suppression : Suppression de l'utilisateur	18	1h30min			
TESTER : Test de la solution	4	0h20min			
EVALUER : Validation de la solution selon le CDC	4	0h20min			
Total tranche	48	4h			
Séquence 18			Date: vendredi, 26 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Rapport	16	1h20min			
Visite - Chef de projet / Experts	2	0h10min	• Visite chef de projet		
DECIDER : Choix de la solution	2	0h10min	• Choix du format du CSV pour la désactivation		
REALISER - Suppression : Désactivation de l'utilisateur	10	0h50min			
TESTER : Test de la solution	4	0h20min			
EVALUER : Validation de la solution selon le CDC	2	0h10min			
Total tranche	38	3h10min			

Séquence 19			Date: mercredi, 31 mai 2023		Matin - 4H00
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Rapport	20	1h40min			
DECIDER : Choix de la solution	2	0h10min			
REALISER - Suppression : Suppression du HomeDirectory	18	1h30min			
TESTER : Test de la solution	4	0h20min			
EVALUER : Validation de la solution selon le CDC	2	0h10min			
Total tranche	48	4h			
Séquence 20			Date: mercredi, 31 mai 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
DECIDER : Choix de la solution	4	0h20min			
REALISER - Suppression : Intégration	12	1h			
TESTER : Test de la solution	7	0h35min			
EVALUER : Validation de la solution selon le CDC	3	0h15min			
DECIDER : Choix de la solution	2	0h10min			
REALISER - Suppression : Génération du CSV	4	0h20min			
TESTER : Test de la solution	2	0h10min			
EVALUER : Validation de la solution selon le CDC	2	0h10min			
Total tranche	38	3h10min			

Séquence 21			Date: vendredi, 2 juin 2023		Matin - 4H00
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Visite - Chef de projet / Experts	14	1h10min	<ul style="list-style-type: none"> • Visite de l'expert • Visite chef de projet 		
Documentation - Rapport	32	2h40min			
Total tranche	48	4h			
Séquence 22			Date: vendredi, 2 juin 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Rapport	36	3h			
Total tranche	38	3h10min			
Séquence 23			Date: lundi, 5 juin 2023		Matin - 3H10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Rapport	36	3h			
Total tranche	38	3h10min			

Séquence 24			Date: lundi, 5 juin 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Rapport	36	3h			
Total tranche	38	3h10min			
Séquence 25			Date: mercredi, 7 juin 2023		Matin - 4H00
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Documentation - Impression et rendu du TPI	46	3h50min			
Total tranche	48	4h			
Séquence 26			Date: mercredi, 7 juin 2023		Aprèm - 3h10
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Documentation - Journal de travail	2	0h10min			
Total tranche	2	0h10min			

5.7 Archives du projet

Lien vers le répertoire Git du projet :

<https://github.com/SylvainPhilipona/TPI-AD-Users-Management>