Gestion des utilisateurs dans l’AD de l’établissement

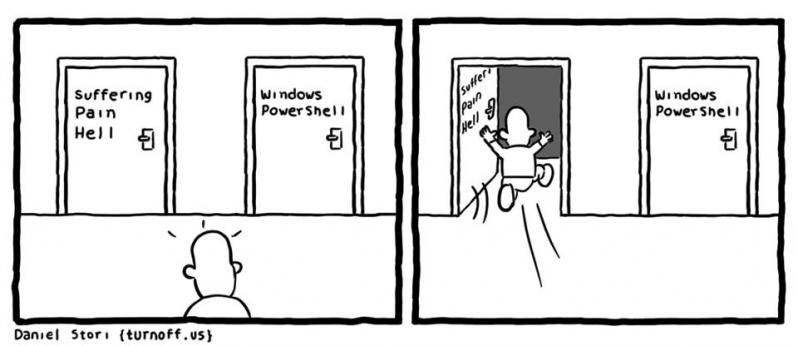


Figure 1 Image de titre

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](#_Toc136440145)

[1.1 Introduction ✔ 3](#_Toc136440146)

[1.2 Objectifs ✔ 4](#_Toc136440147)

[1.3 Planification initiale 5](#_Toc136440148)

[2 Analyse / Conception 7](#_Toc136440149)

[2.1 Concept 7](#_Toc136440150)

[2.2 Outil de versioning ✔ 7](#_Toc136440151)

[2.3 Stratégie de test 7](#_Toc136440152)

[2.4 Risques techniques 9](#_Toc136440153)

[2.5 Planification 10](#_Toc136440154)

[2.6 Dossier de conception 11](#_Toc136440155)

[2.6.1 Création des comptes ✔ 11](#_Toc136440156)

[2.6.2 Modification des comptes ✔ 12](#_Toc136440157)

[2.6.3 Suppression des comptes 13](#_Toc136440158)

[3 Réalisation 15](#_Toc136440159)

[3.1 Dossier de réalisation 15](#_Toc136440160)

[3.1.1 Environnement ✔ 15](#_Toc136440161)

[3.1.2 Version du projet ✔ 15](#_Toc136440162)

[3.1.3 Normes de codage et de nommage 15](#_Toc136440163)

[3.1.4 Création des comptes 18](#_Toc136440164)

[3.2 Description des tests effectués 29](#_Toc136440165)

[3.3 Erreurs restantes 29](#_Toc136440166)

[3.4 Liste des documents fournis 29](#_Toc136440167)

[3.5 Glossaire 29](#_Toc136440168)

[4 Conclusions 30](#_Toc136440169)

[5 Annexes 31](#_Toc136440170)

[5.1 Résumé du rapport du TPI / version succincte de la documentation 31](#_Toc136440171)

[5.2 Sources – Bibliographie 31](#_Toc136440172)

[5.3 Journal de travail 31](#_Toc136440173)

[5.4 Manuel d'Installation 31](#_Toc136440174)

[5.5 Manuel d'Utilisation 31](#_Toc136440175)

[5.6 Archives du projet 31](#_Toc136440176)

# Analyse préliminaire

## 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

## 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.

## Planification initiale

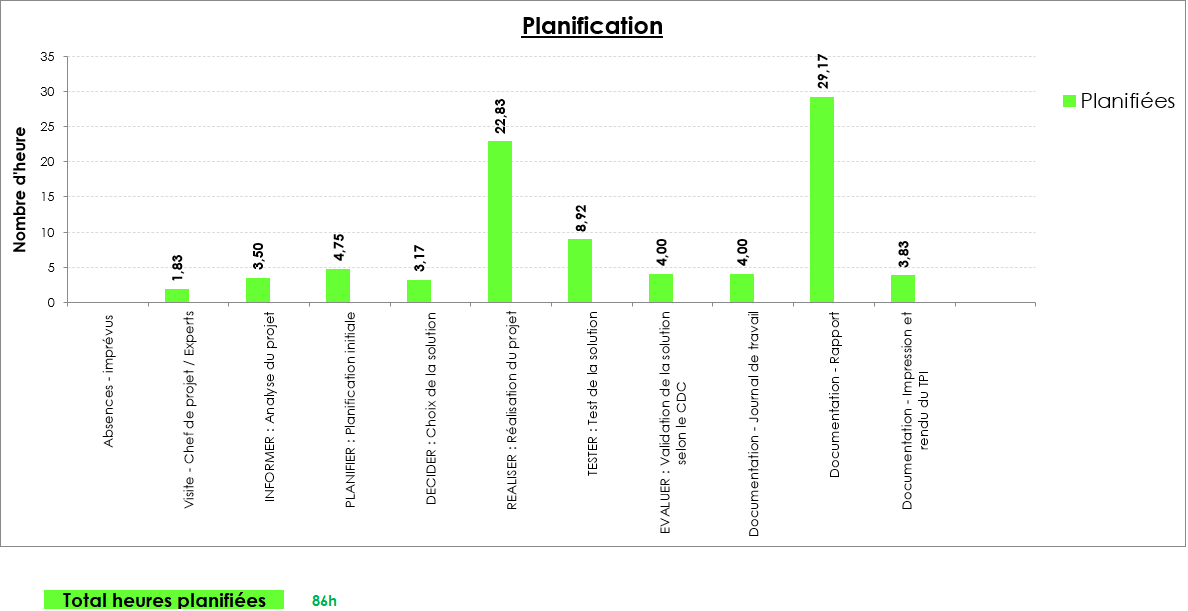


Figure 2 Planification initiale

La planification initiale du projet a été élaborée en suivant méthode des 6 pas.

86H => Car pense finir en avance

*Ce chapitre montre la planification du projet. Celui-ci peut être découpé en tâches qui seront planifiées. Il s'agit de la première planification du projet, celle-ci devra être revue après l'analyse. Cette planification sera présentée sous la forme d'un diagramme.*

*Ces éléments peuvent être repris des spécifications de départ.*

# Analyse / Conception

## Concept

Le concept de ce projet se résume en l’automatisation de la création, la modification et la suppression des comptes utilisateurs. Ces comptes sont utilisés par les étudiants et apprentis pour se connecter aux ordinateurs du domaine.

*Le concept complet avec toutes ses annexes:*

*Par exemple :*

* *Multimédia: carte de site, maquettes papier, story board préliminaire, …*
* *Bases de données: interfaces graphiques, modèle conceptuel.*
* *Programmation: interfaces graphiques, maquettes, analyse fonctionnelle…*
* *…*

## 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".

## Stratégie de test

|  |  |
| --- | --- |
| Test | Impact en cas d’échec |
| Une recherche est effectuée pour vérifier si l'utilisateur existe déjà dans l'AD. | Impossibilité de vérifier si un compte existe déjà. Cela peut entrainer une réinitialisation non voulue du compte d’un apprenti. |
| 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. |  |
| Le nom d’ouverture de session proposé est unique, dans le cas contraire un autre est généré. | Problèmes dans le cas de personnes ayant le même prénom et nom de famille. Le nom de d’ouverture de session devant être unique. |
| Un mot de passe aléatoire devant être changé à l’ouverture de session est généré. | L’utilisateur ne peut pas changer de mot de passe. |
| Un script d’ouverture de session est généré selon la profession de l’utilisateur. |  |
| Une description est générée selon la classe de l’utilisateur. |  |
| L’utilisateur est créé dans la bonne OU. |  |
| Le dossier personnel est créé avec les bons droits d’accès. | Soit impossibilité d’avoir un dossier personnel, ou bien risque de compromission des données personnelles si les droits d’accès sont erronés. |
| L’utilisateur est membre des groupes spécifiés dans le CSV. |  |
| Un fichier CSV contenant les actions effectuées est généré. |  |

|  |  |
| --- | --- |
| Test | Impact en cas d’échec |
| Une recherche par le nom d’ouverture de session est effectuée pour vérifier si l'utilisateur existe déjà dans l'AD. |  |
| Le compte est modifié selon les informations du CSV. |  |
| Un fichier CSV contenant les modifications effectuées est généré. |  |

|  |  |
| --- | --- |
| Test | Impact en cas d’échec |
| Une recherche par le nom d’ouverture de session est effectuée pour vérifier si l'utilisateur existe déjà dans l'AD. |  |
| Il est possible de choisir de désactiver ou supprimer un compte. |  |
| Un compte est supprimé uniquement s’il est désactivé. |  |
| Le dossier personnel est supprimé lors de la suppression du compte. |  |
| Un fichier CSV contenant les modifications et suppressions est généré. |  |

*Décrire la stratégie globale de test:*

* *types de des tests et ordre dans lequel ils seront effectués.*
* *les moyens à mettre en œuvre.*
* *couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).*
* *données de test à prévoir (données réelles ?).*
* *les testeurs extérieurs éventuels.*

## Risques techniques

* *risques techniques (complexité, manque de compétences, …).*

*Décrire aussi quelles solutions ont été appliquées pour réduire les risques (priorités, formation, actions, …).*

## Planification

*Révision de la planification initiale du projet :*

* *planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.*
* *partage des tâches en cas de travail à plusieurs.*

*Il s’agit en principe de la planification* ***définitive du projet****. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l’historique.*

## Dossier de conception

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

### 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 remplis, 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.

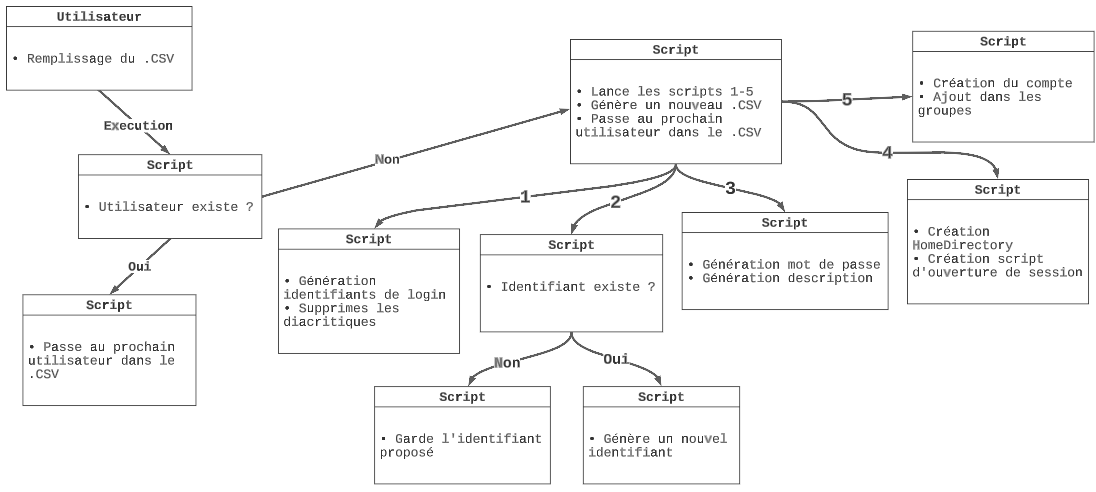


Figure 3 Schéma du processus de création des comptes

### 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 remplis, 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 déjà. Si c’est le cas, le script passe à l’apprenti suivant. Dans le cas contraire, la phase de modification du compte commence.
* 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 modifiés dans le compte. Ensuite un nouveau fichier CSV est généré avec les modifications effectuées.

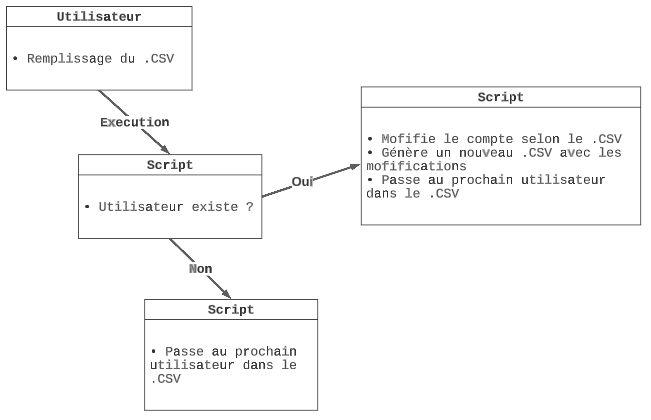


Figure 4 Schéma du processus de modification des comptes

### Suppression des comptes

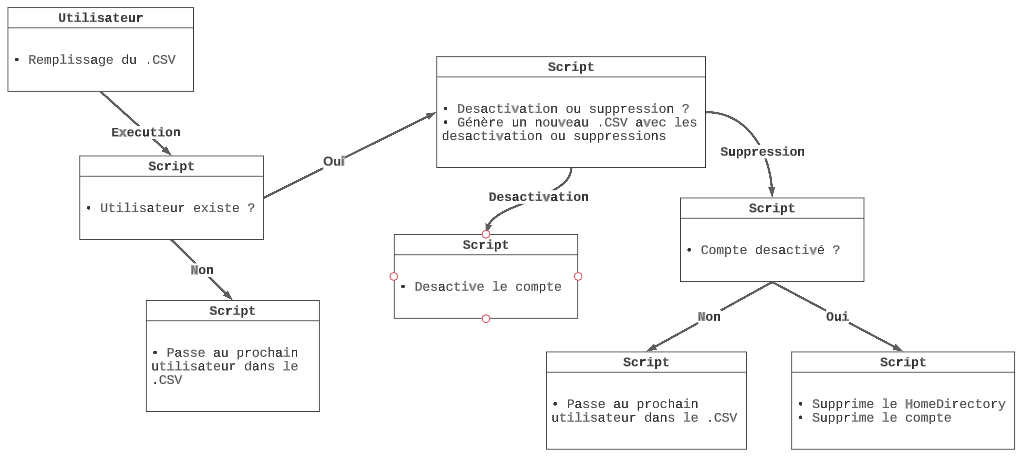


Figure 5 Schéma du processus de suppression des comptes

*Fournir tous les document de conception:*

* *le choix du matériel HW*
* *le choix des systèmes d'exploitation pour la réalisation et l'utilisation*
* *le choix des outils logiciels pour la réalisation et l'utilisation*
* *site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, …*
* *bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.*
* *programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme…*

***Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !***

# Réalisation

## Dossier de réalisation

### 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
* PowerShell, Version 5.1

### Version du projet ✔

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

### Normes de codage et de nommage

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

#### 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.



Figure 6 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.



Figure 7 Format de nommage de paramètres

#### 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.



Figure 8 Format de nommage de constantes

#### 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>

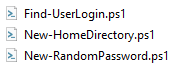


Figure 9 Format de nommage des scripts

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>

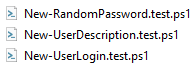


Figure 10 Format de nommage des tests unitaires

#### 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.



Figure 11 Format d'entête de scripts

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

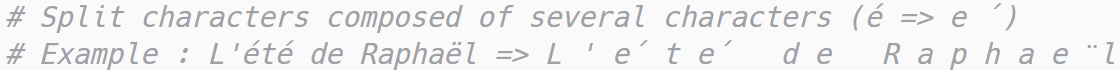


Figure 12 Format de commentaires

### Création des comptes

#### 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**

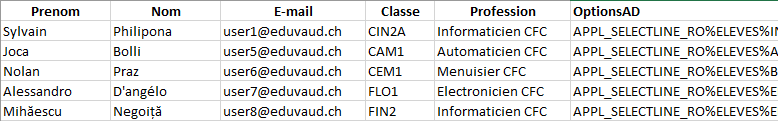


Figure 13 Exemple de données dans le CSV

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 chaine 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.



Figure 14 Division des groupes avec le caractère "%"

#### Exécution du script

Une fois le fichier CSV remplis 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 crée 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.

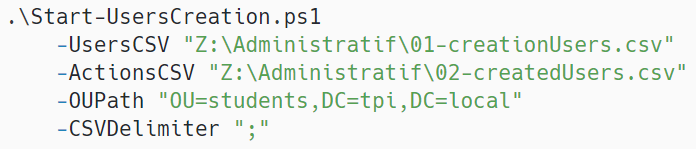


Figure 15 Exemple d'exécution du script de création avec des paramètres

#### 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 :

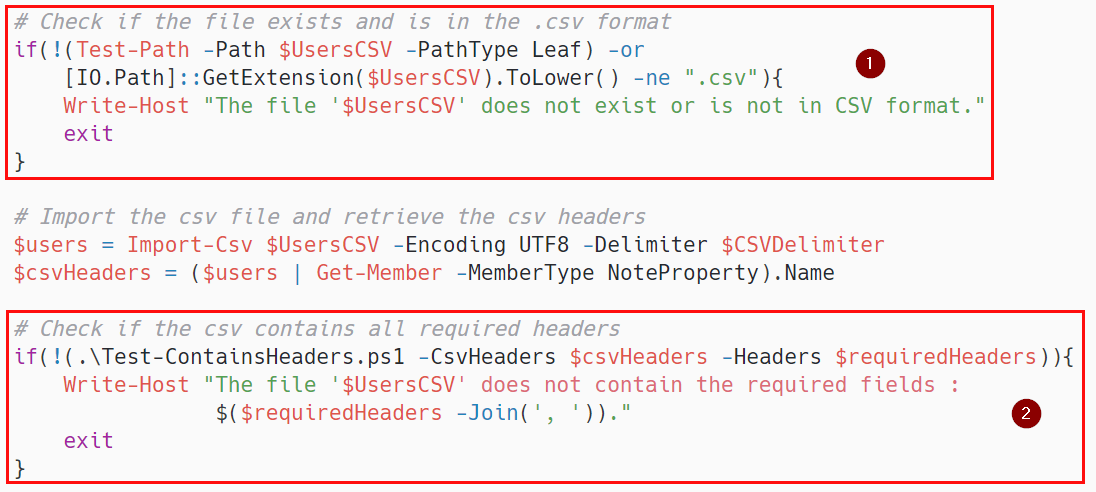


Figure 16 Vérification des données du fichier CSV

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’errer est affiché à l’utilisateur et l’exécution du script s’arrête.

1. **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’errer est affiché à l’utilisateur et l’exécution du script s’arrête.

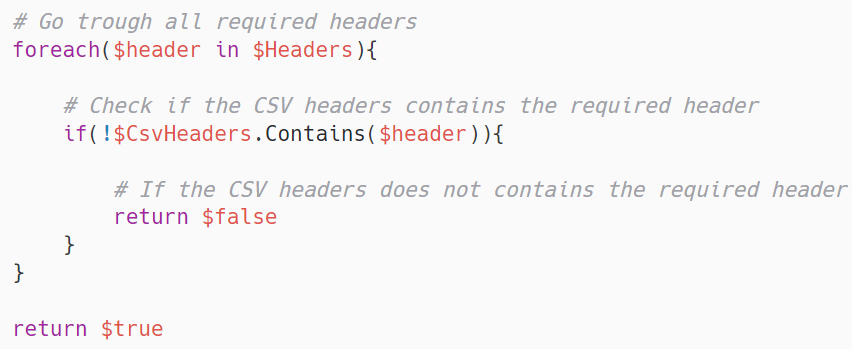


Figure 17 Vérification des en-têtes du fichier

#### 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.

##### 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.



Figure 18 Recherche d'un utilisateur par son nom et prénom

##### Génération de l’identifiant de connexion

Le nom de l’identifiant de connexion doit est 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.

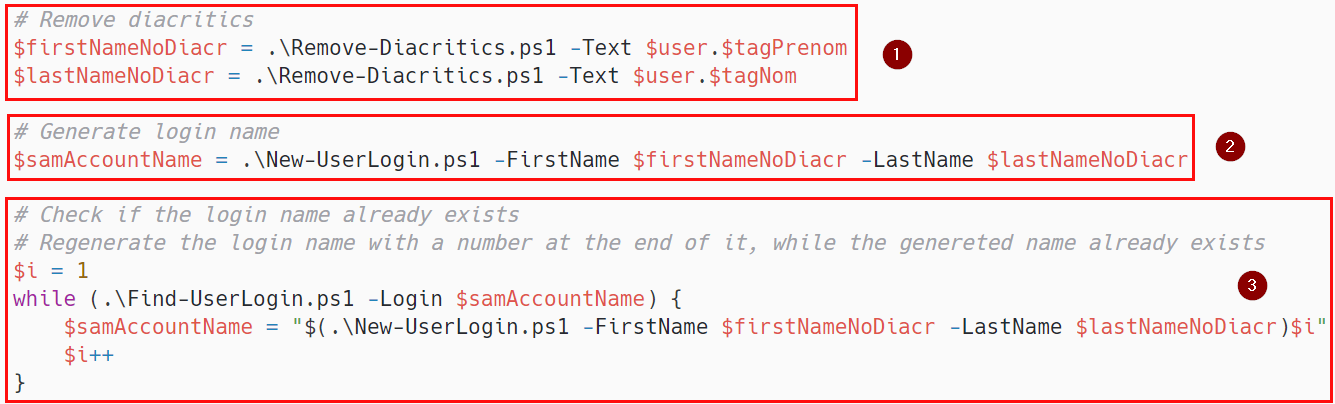


Figure 19 Génération de l’identifiant de connexion

1. **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 êtres 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 chaine 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 chaine 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.

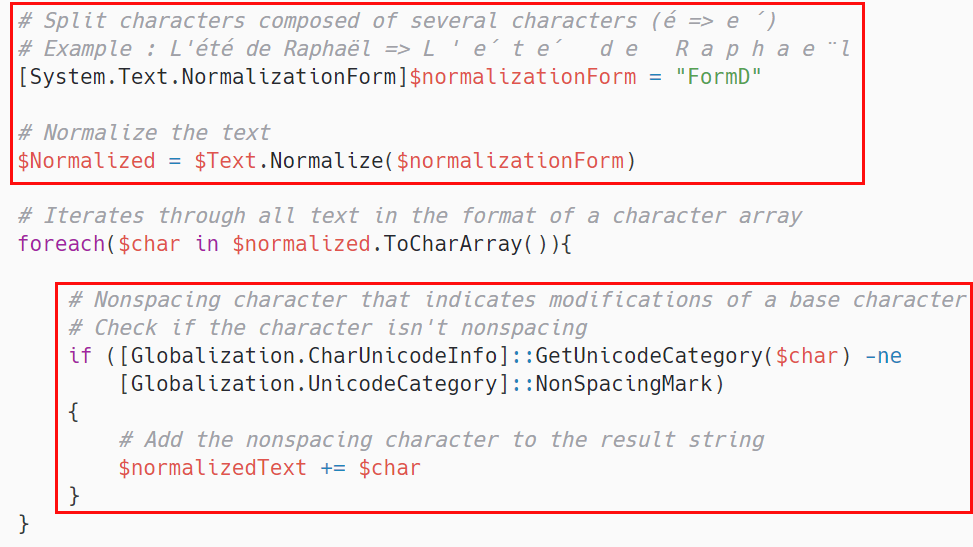


Figure 20 Suppression des caractères diacritiques

1. **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.

1. **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

##### 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.

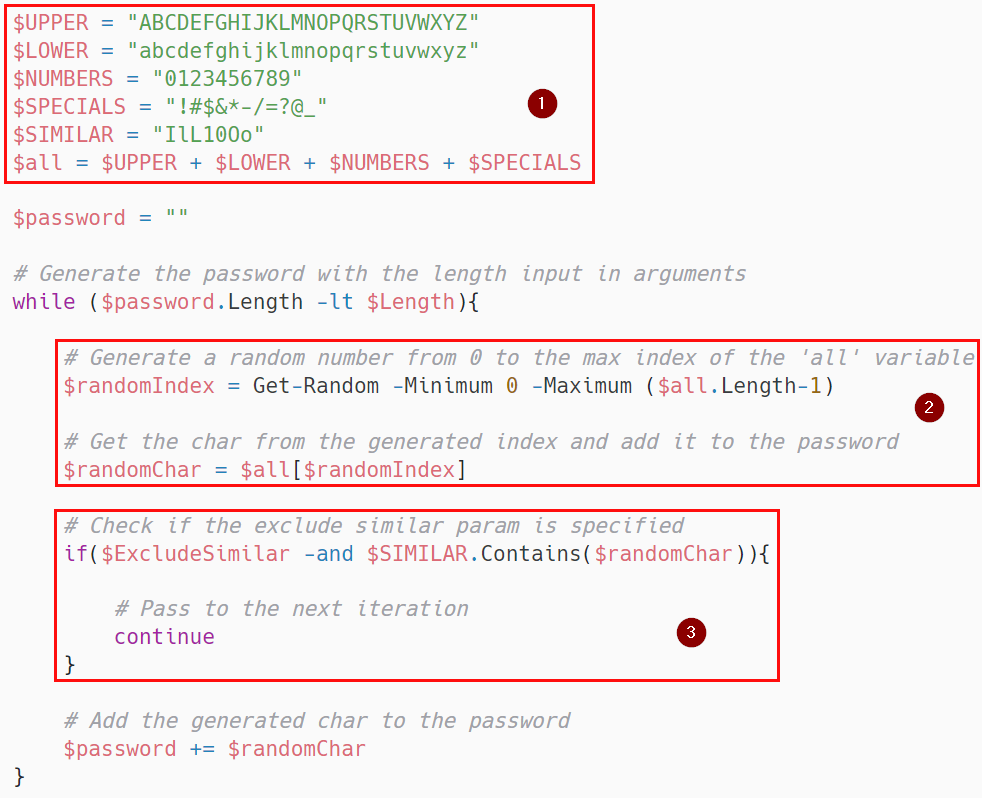


Figure 21 Génération du mot de passe

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.

1. **Génération aléatoire d’un caractère**

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

1. **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 êtres 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.

##### 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.

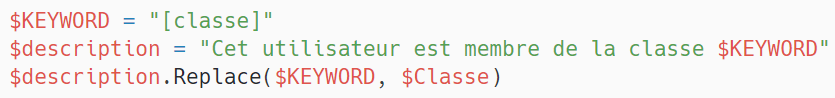


Figure 22 Génération de la description

##### 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.



Figure 23 Génération du script de connexion

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.

1. **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.

##### Création du compte

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



Figure 24 Création du compte

1. **Définition des données utilisateur**

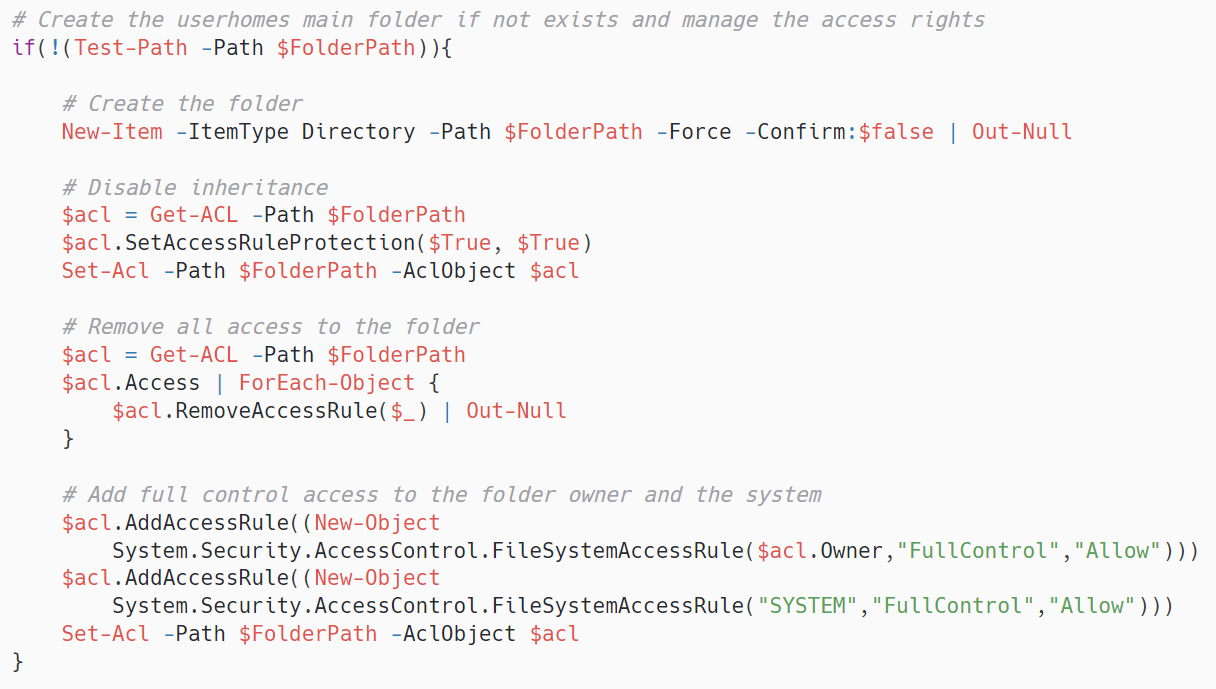
Un dictionnaire est créé contenant toutes les données nécessaires à la création de l’utilisateur.

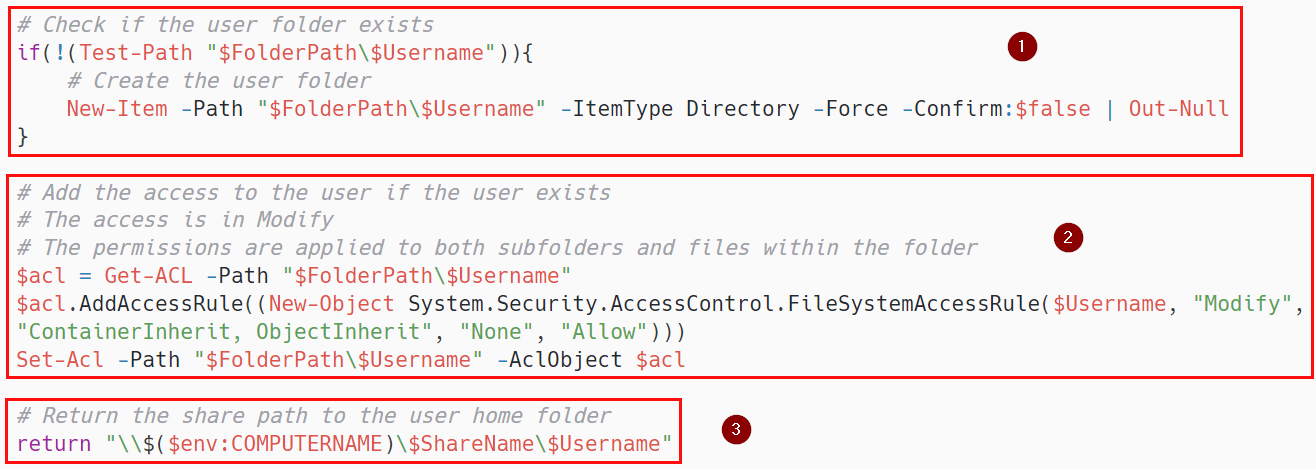
1. **Création et activation du compte**

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.

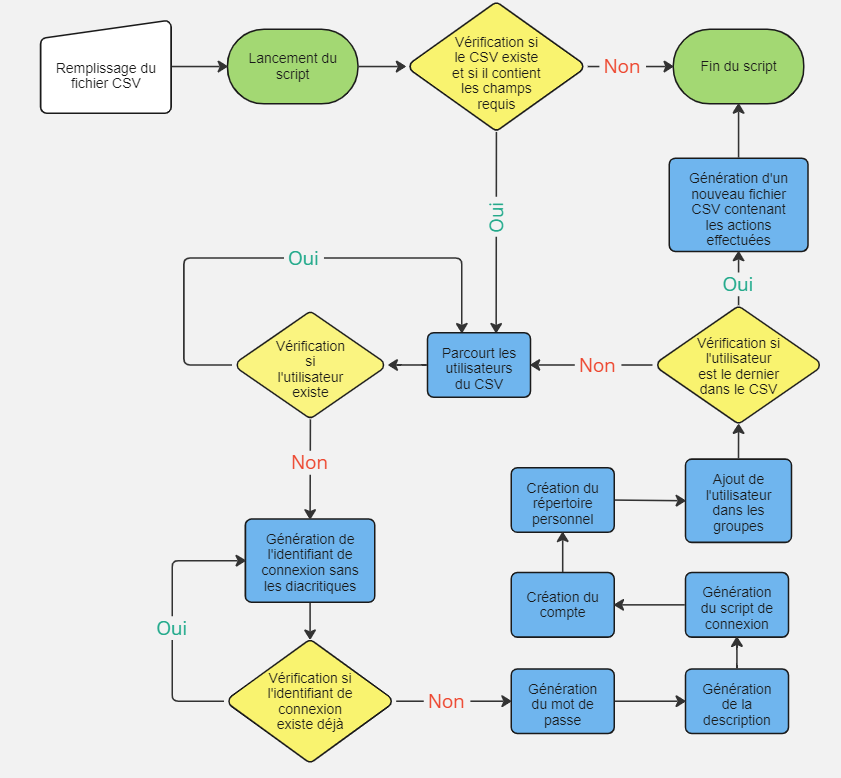
##### Création du répertoire personnel





##### Ajout dans les groupes

#### Génération du CSV



*Décrire la réalisation "physique" de votre projet*

* *les répertoires où le logiciel est installé*
* *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
* *les versions des systèmes d'exploitation et des outils logiciels*
* *la description exacte du matériel*
* *le numéro de version de votre produit !*
* *programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

*NOTE : Evitez d’inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n’incluez Fue cette partie…*

## Description des tests effectués

*Pour chaque partie testée de votre projet, il faut décrire:*

* *les conditions exactes de chaque test*
* *les preuves de test (papier ou fichier)*
* *tests sans preuve: fournir au moins une description*

## Erreurs restantes

*S'il reste encore des erreurs:*

* *Description détaillée*
* *Conséquences sur l'utilisation du produit*
* *Actions envisagées ou possibles*

## Liste des documents fournis

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

* *le rapport de projet*
* *le manuel d'Installation (en annexe)*
* *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
* *autres…*

## Glossaire

1. Active Directory (AD)
2. Annuaire
3. Scripting
4. Domaine
5. CSV
6. Diacritiques
7. Merged
8. OU
9. Tests unitaires
10. Framework
11. Forme de normalisation
12. Fonction
13. Itération
14. Valeur null

# Conclusions

*Développez en tous cas les points suivants:*

* *Objectifs atteints / non-atteints*
* *Points positifs / négatifs*
* *Difficultés particulières*
* *Suites possibles pour le projet (évolutions & améliorations)*

# Annexes

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

## Sources – Bibliographie

*Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)… Et de toutes les aides externes (noms)*

## Journal de travail

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Durée** | **Activité** | **Remarques** |
|  |  |  |  |
|  |  |  |  |

## 

## Manuel d'Installation

## Manuel d'Utilisation

## Archives du projet

*Media, … dans une fourre en plastique*