

# Visual Basic pour Applications

TP

20/03/2020

**LE FEYER** Aymeric

## Table des matières

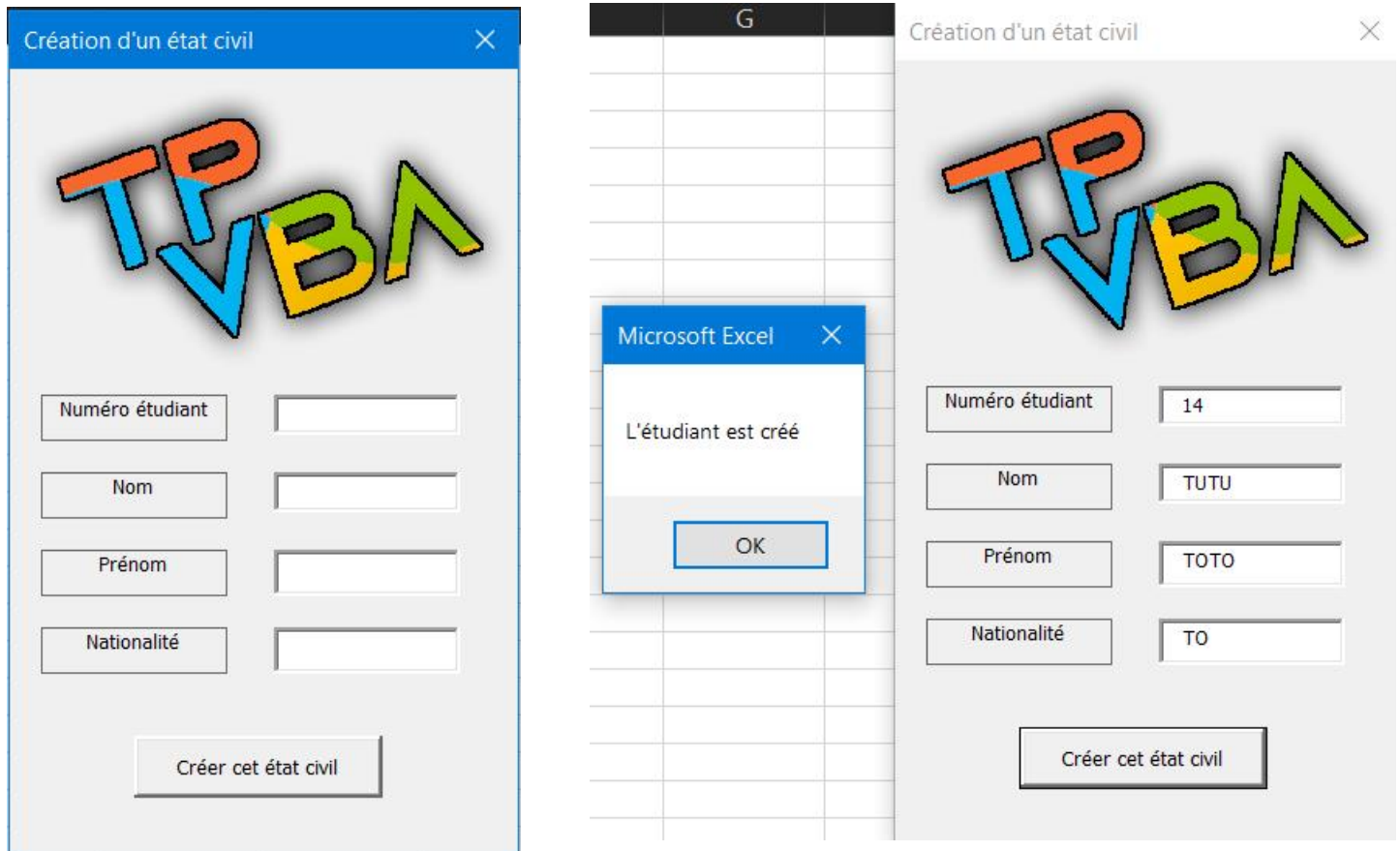
Exercice 1 – Création de l'état civil d'un étudiant .....	2
Captures d'écrans.....	2
Analyse du code.....	2
Exercice 2 – Affichage de l'état civil.....	2
Captures d'écrans.....	3
Analyse du code.....	3
Exercice 3 – Gestion des notes et adresses .....	3
Captures d'écrans.....	3
Analyse du code.....	4
Création .....	4
Affichage .....	4
Exercice 4 – Modifications.....	4
Captures d'écrans.....	4
Analyse du code.....	5
Exercice 5 – Requêtes.....	6
Captures d'écrans.....	6
Analyse du code.....	7

**TOUS LES CODES SONT COMMENTES**

## Exercice 1 – Création de l'état civil d'un étudiant

Vous retrouverez le code commenté dans le fichier **Creation.frm**

### Captures d'écrans



### Analyse du code

On commence par effectuer quelques tests sur les valeurs.

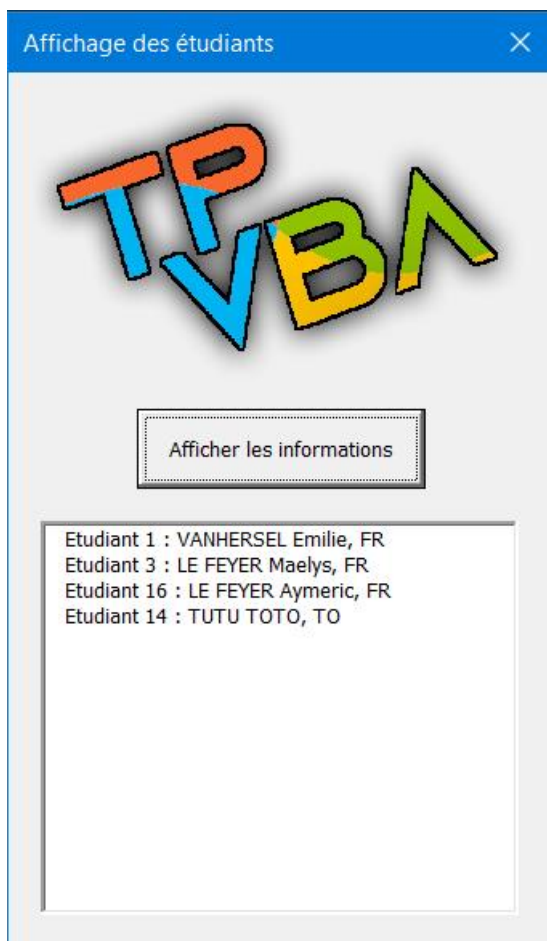
Si tous les tests sont réussis, on ajoute ouvre le fichier Etatcivil en mode **Append** pour ajouter une ligne avec les informations de l'étudiant

On utilise une fonction tierce pour compter le nombre de virgule dans les champs  
La virgule étant le séparateur de valeurs dans mes documents, je l'interdis

## Exercice 2 – Affichage de l'état civil

Vous retrouverez le code commenté dans le fichier **Affichage.frm**

## Captures d'écrans



### Analyse du code

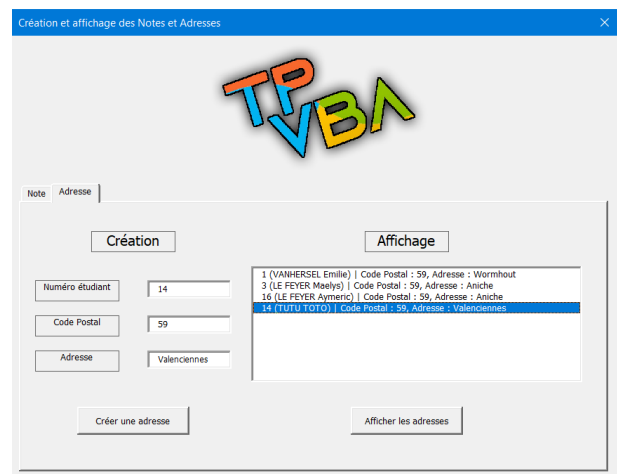
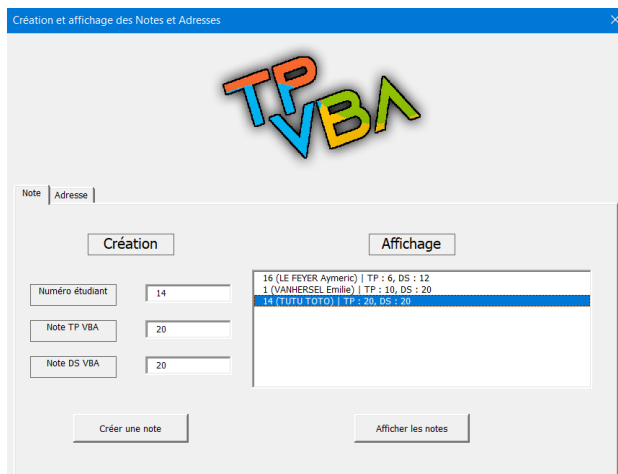
L'affichage est simple et sera utilisé de nombreuses reprises durant le TP. On ouvre un fichier, pour chaque ligne, on utilise la fonction Split pour séparer les lignes par leurs virgules, afin d'avoir un format d'affichage plus propre.

Dans notre cas, pour chaque ligne, on ajoute à la ListBox les éléments formatés.

## Exercice 3 – Gestion des notes et adresses

Vous retrouverez le code commenté dans le fichier **NoteAdresse.frm**

### Captures d'écrans



## Analyse du code

Les notes et les adresses fonctionnent de la même manière

### Création

On commence par vérifier que la note (ou l'adresse) n'existe pas déjà pour le numéro d'étudiant indiqué

Si c'est bon, on parcourt la liste des états civils pour vérifier que le numéro d'étudiant existe

Ensuite on ajoute la ligne dans le fichier correspondant, en vérifiant qu'aucun champ n'est vide et que les notes soient numériques

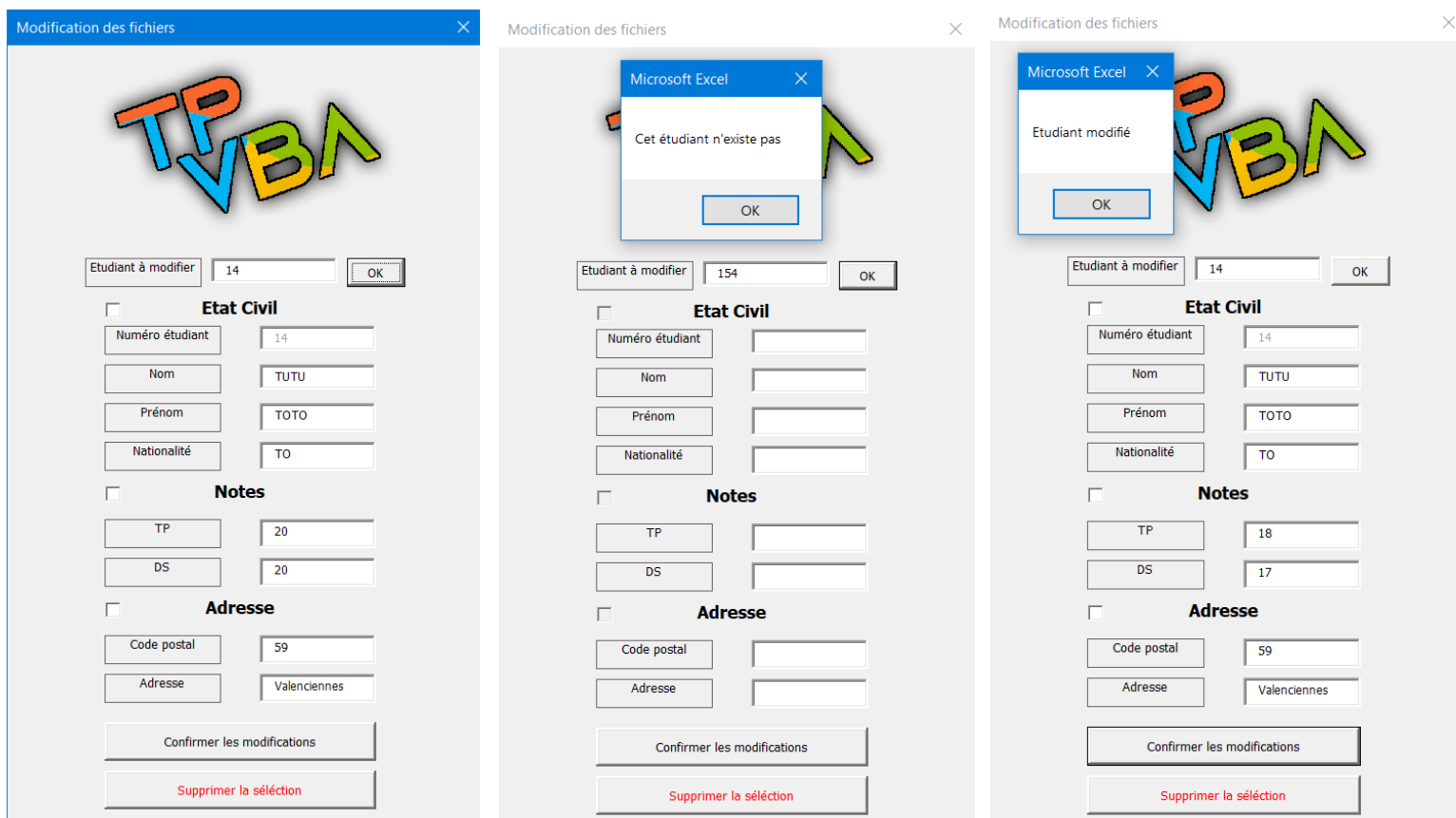
### Affichage

L'affichage se passe comment l'exercice précédent


On ouvre deux fichiers afin d'avoir un formatage de donnée propre, et d'avoir pour chaque note et adresse, le nom et le prénom de l'étudiant (en plus de son numéro)

## Exercice 4 – Modifications

### Captures d'écrans



Modification des fichiers



Etudiant à modifier : 14 OK

☐ **Etat Civil**

Número étudiant : 14

Nom : TUTU

Prénom : TOTO

Nationalité : TO

☒ **Notes**

TP :

DS :

☒ **Adresse**

Code postal :

Adresse :

Confirmer les modifications

Supprimer la sélection

On peut supprimer des informations avec les CheckBox

### Analyse du code

Le code étant commenté, et n'ayant plus beaucoup de place sur mon compte rendu, je vais condenser les analyses

Concrètement, on fait des jointures de fichiers avec comme clé primaire le numéro de l'étudiant

On affiche ensuite toutes les informations dans les différentes TextBox

L'utilisateur pourra modifier directement les cases puis confirmer les modifications

Il pourra aussi s'il le souhaite, cocher l'une des trois (ou les trois) cases afin de supprimer les éléments.

Tout un tas de tests conditionnels est vérifié afin de ne pas rentrer en conflit avec d'autres formats de fichiers

Tout est modifiable, sauf le numéro de l'étudiant.

Si on supprime l'état civil, on supprimera aussi forcément les notes et adresses, qui ne peuvent plus exister

Modifier une valeur revient à supprimer tout le fichier et le remplir avec les valeurs modifiées

# Exercice 5 – Requêtes

## Captures d'écrans

The screenshots show a VBA application with a menu bar: Ville donnée, Nom donné, Moyenne VBA, Suppression par prénom, Fusion, Affichage de la fusion. The application has a logo 'TP VBA' and a title bar 'Tout un tas de requêtes !'.

**Screenshot 1:** The 'Ville donnée' menu item is selected. The title is 'Lister tous les noms des étudiants qui habitent une ville donnée'. The input fields are 'Ville' and 'Aniche', with an 'OK' button. The output box shows: LE FEYER Maelys, LE FEYER Aymeric.

**Screenshot 2:** The 'Nom donné' menu item is selected. The title is 'Lister les données des étudiants (numéro, nom, prénom, ville) qui ont le même nom donné'. The input field is 'Nom' with the value 'LE FEYER' and an 'OK' button. The output box shows: 3 - LE FEYER Maelys - Aniche, 16 - LE FEYER Aymeric - Aniche.

**Screenshot 3:** The 'Moyenne VBA' menu item is selected. The title is 'Donner la moyenne de Programmation sous Excel d'un étudiant donné à partir de son nom et prénom (la note de TP est coefficient 1 et la note d'examen 2)'. The input fields are 'Nom' (LE FEYER), 'Prénom' (Aymeric), and 'Moyenne' (10/20), with an 'OK' button.

**Screenshot 4:** The 'Suppression par prénom' menu item is selected. The title is 'Supprimer tous les étudiants dont le prénom commence par une lettre donnée'. The input field is 'Lettre' with the value 'T' and an 'OK' button. The output box shows: Liste des étudiants, Etudiant 1 : VANHERSEL Emilie, FR, Etudiant 3 : LE FEYER Maelys, FR, Etudiant 16 : LE FEYER Aymeric, FR. There is an 'Actualiser' button at the bottom.

Toto est supprimé

## Analyse du code

**Ville donnée** On parcourt les états civils et les adresses, on fait la jointure sur le numéro de l'étudiant et on affiche tous les étudiants quand la ville correspond

**Nom donné** On a besoin d'effectuer une jointure pour pouvoir afficher l'adresse de la personne. On fait cela avec des tableaux globaux comme depuis le début du TP

**Moyenne VBA** On récupère les notes, on effectue un calcul de Moyenne et on affiche le tout dans une TextBox disabled. Une jointure est encore une fois faite

**Suppression par prénom** On utilise le même principe que depuis le début, on ajoute dans un tableau tout ce qu'on veut garder, soit tous les étudiants dont le prénom ne commence pas par la lettre. Puis on écrase le fichier en ajoutant le tableau

**Fusion** Voici la requête la plus intéressante. On commence par cocher toutes les valeurs à fusionner. On récupère ces valeurs dans le code. On effectue des jointures par numéro étudiant, et pour chaque valeur, on teste si l'utilisateur souhaite ou non la fusionner, puis on l'ajoute au tableau. A la fin, on écrase le fichier Fusion en écriture pour y mettre toutes les lignes du tableau.

Finalement, on vide le tableau pour ne pas le surcharger, et pour être prêt à la prochaine fusion