

Cahier des charges

Nutrition Bourse



Table des matières

Introduction	3
Spécifications Fonctionnelles	3-4
Déroulement de l'application	4-5
Maquette nutrition Bourse	5-6
Rectifications fonctionnalités	7
Récapitulatif des étapes de codes	8-21
Conclusion	21

1) Introduction :

Le présent cahier des charges définit les objectifs, les spécifications et les exigences pour la conception et la mise en œuvre de l'application "Nutrition Bourse". L'objectif de ce projet est de créer une application VBA dédiée à la nutrition pour faciliter la gestion des apports nutritionnels.

L'application "Nutrition Bourse" sera principalement développée en utilisant des "UserForm" via Visual Basic for Applications (VBA) dans Microsoft Excel.

Problématique Principale :

Permettre aux utilisateurs de calculer rapidement et précisément les apports nutritionnels de leurs repas en fonction des aliments sélectionnés.

Problématique secondaire :

Intégrer une base de données complète des apports nutritionnels des produits couramment consommés.

Permettre aux utilisateurs de déterminer les quantités nécessaires pour atteindre des objectifs nutritionnels spécifiques.

2) Spécifications Fonctionnelles :

- L'application "Nutrition Bourse" doit comporter les fonctionnalités suivantes :
 1. Sélection d'Aliments
 - Permettre à l'utilisateur de sélectionner les aliments qu'il souhaite inclure dans son repas.
 - Le choix des aliments se fera par un simple clic sur l'image du produit.
 2. Calcul des Apports Nutritionnels

Calculer instantanément les apports nutritionnels totaux du repas en fonction des aliments sélectionnés. En prenant en compte l'entrée, le plat et le dessert. Afficher les résultats de manière claire et compréhensible pour l'utilisateur.

3. Historique des repas

Garder les repas déjà fait part les utilisateurs sur une seconde page. Qui permet de revoir les valeurs nutritionnelles des repas précédents.

4. Base de Données des Apports Nutritionnels

Intégrer une base de données complète comprenant les apports nutritionnels (calories, protéines, glucides, lipides, vitamines, minéraux, etc.) de produits courants.

3) Déroulement de l'Application :

L'utilisateur débutera l'application dans Microsoft Excel, où il trouvera une interface simple comportant un seul bouton intitulé "Compose ton repas." Après avoir cliqué sur ce bouton, le processus se décomposera en cinq phases :

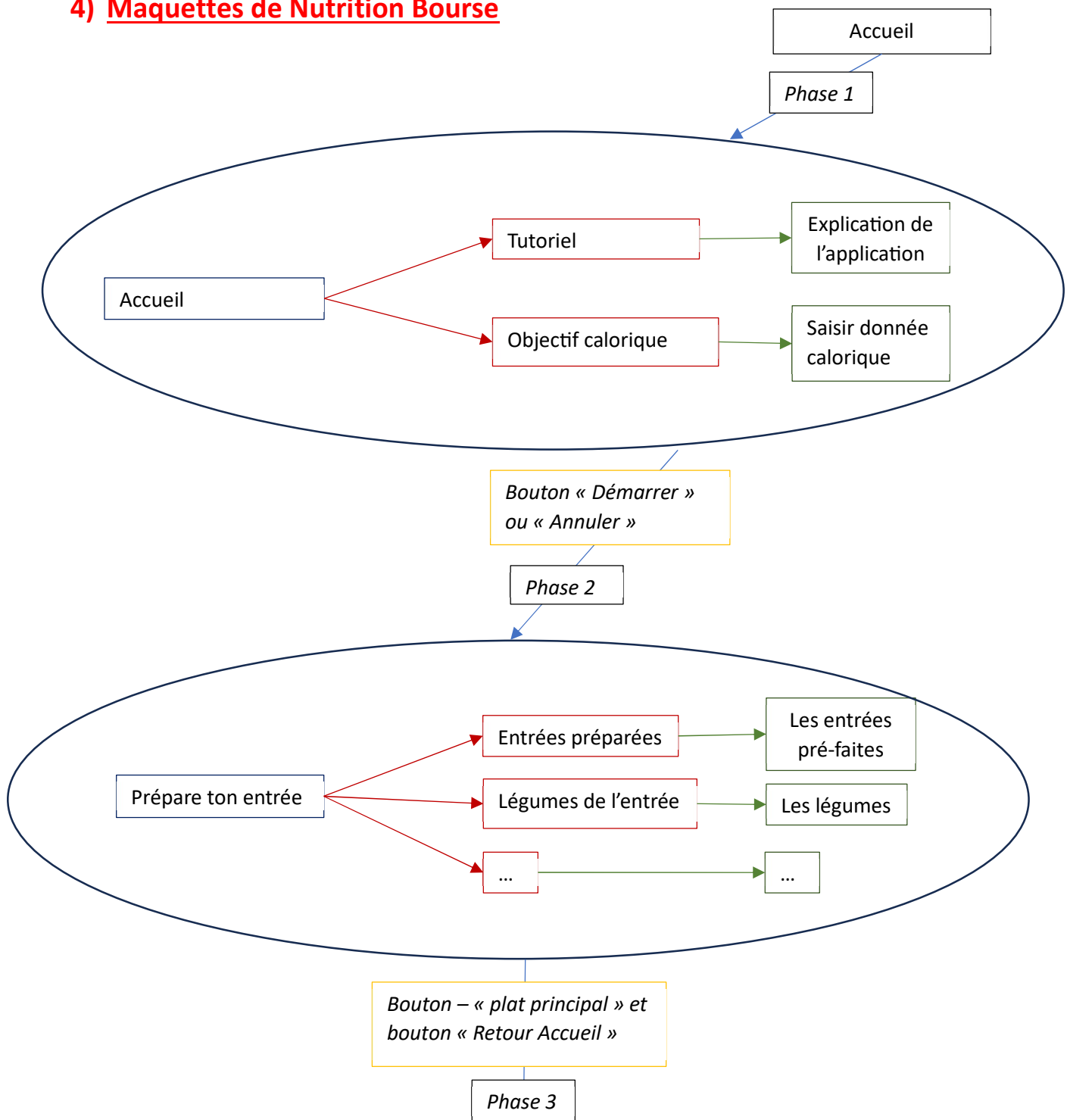
Chaque étape de sélection (Prépare ton entrée – Prépare ton plat – Prépare ton dessert) prend une page entière.

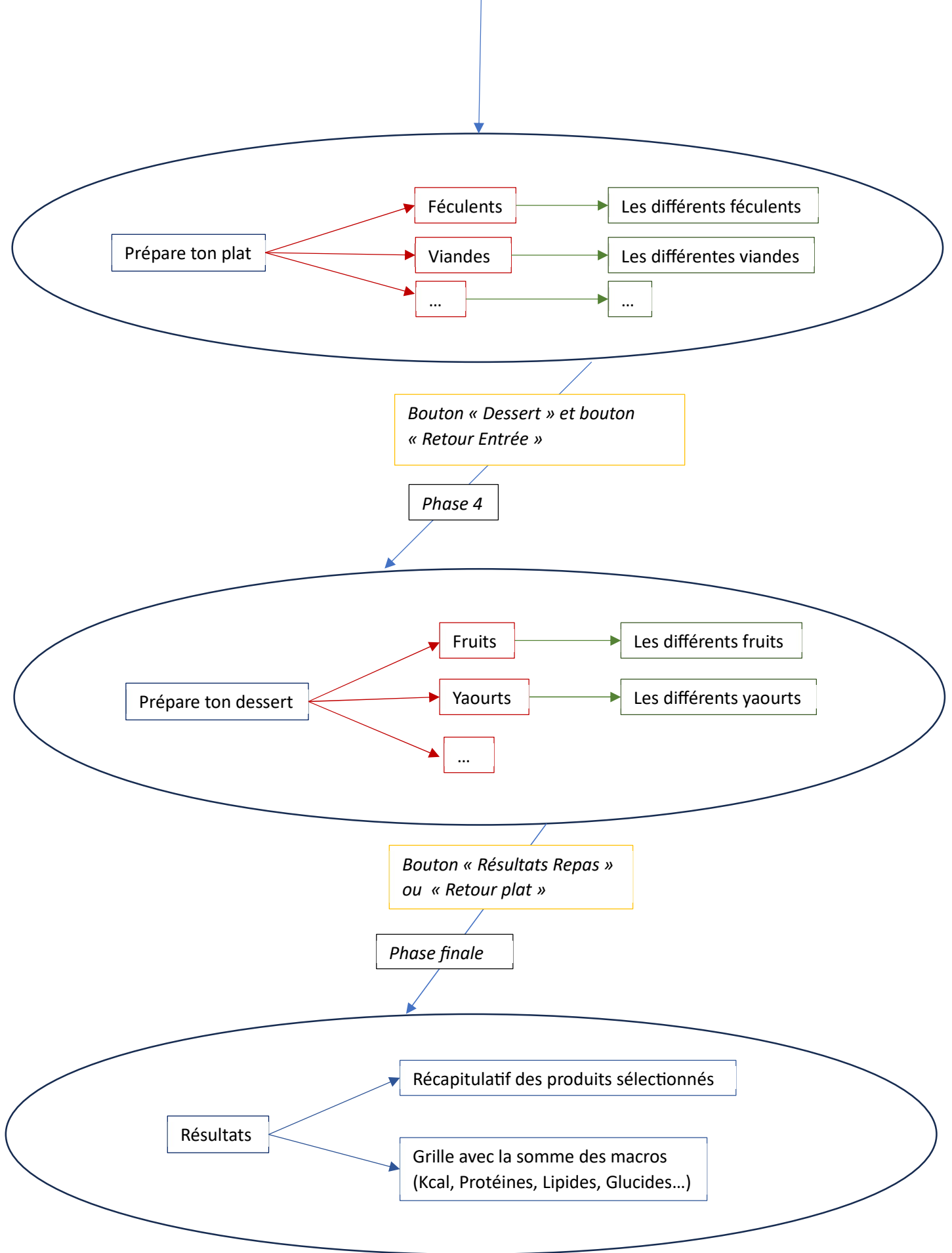
- Ouverture de l'Application : Cette première étape s'ouvre dans Excel et présente l'interface avec le bouton "Accueil."
- Phase 1 (Accession à l'accueil) : lorsque l'utilisateur clique sur le bouton accueil, il accède à une page proposant un tutoriel détaillé de l'utilisation de l'application et de la capacité à appliquer une sélection pour un objectif calorique pour son repas.
- Phase 2 (Prépare ton entrée) : Dans cette phase, l'utilisateur sélectionne son entrée parmi plusieurs sous-catégories, chacune contenant une unique catégorie, telle que "Entrées préparées." Chaque produit sera accompagné d'une image associée, cliquable pour l'ajouter à un panier visible à tout moment.
- Phase 3 (Prépare ton plat) : Cette phase fonctionne de manière similaire à la Phase 1, mais se concentrera sur le plat principal.
- Phase 4 (Prépare ton dessert) : De nouveau, similaire aux phases précédentes, cette étape portera sur les desserts.
- Phase Finale (Résultat) : Dans cette dernière phase, l'application présentera un récapitulatif des produits choisis. Les résultats seront affichés sous forme de grille, indiquant la somme des apports nutritionnels (Kcal, Protéines, Lipides, Glucides, etc.) et ayant la capacité à la fin de pouvoir mémoriser son repas.

L'application "Nutrition Bourse" a été conçue pour offrir les fonctionnalités suivantes :

- Une progression linéaire à travers les phases, mais avec la flexibilité de passer à la phase suivante sans avoir sélectionné de produits ou exploré les sous-catégories.
- La possibilité de sélectionner autant de produits que souhaité.
- Des boutons "suivant" et "précédent" disponibles à chaque étape, pour faciliter la navigation dans le processus.
- Il sera possible sur une autre page d'avoir un historique des repas fait par l'utilisateur.

4) Maquettes de Nutrition Bourse





5) Rectification des fonctionnalités

1- Éléments supprimés au cours du projet

- Option de recherche : le manque de temps et la difficulté de l'écriture de ce code ne permettait pas de pouvoir le réaliser de manière parfaite et efficace dans les temps requis. Ainsi cette option n'aurait pas été réellement agréable pour l'utilisateur.
- Bulle d'information nutritionnelle pour chaque aliment : au cours de la réalisation du projet nous n'y avons plus vu d'intérêt de l'effectuer car lorsque l'on clique sur l'aliment, la valeur nutritionnelle pour 100g s'affiche automatiquement pour chaque produit.
- Proposition de combinaison d'aliment : cet élément n'a pas été rajouté car celui-ci n'était pas notre priorité du fait que cela aurait rajouté des combinaisons de plats qui ne composent pas le cœur du projet et aurait rajouté des écritures sans de réelles nouveautés.
- Les recherches des différentes valeurs nutritionnelles : ont été réalisées sur des sites en ligne tels que « La nutrition » (lien : <https://www.lanutrition.fr/les-aliments-a-la-loupe>), ou encore l'utilisation de ChatGPT, nous permettant de gagner du temps et d'avoir un échantillon de valeur provenant d'une seule source. Ainsi, cela permet d'avoir des produits évalués sur une même base, offrant la vision la plus identique des apports nutritionnels des produits.

2- Éléments rajoutés au cours du projet

- Changement de nom des commandes boutons : En effet, les boutons ont globalement changé de nom par rapport aux noms donnés initialement pour une meilleure compréhension de transition vers un autre userform dans l'application.
- Apport d'une nouvelle étape nommée « Accueil » : Le nom d'accès comme nous l'avons initié dans le point précédent, a changé pour « Accueil ». Cet ajout d'une nouvelle phase aux prémices de l'application permet à l'utilisateur de pouvoir mieux se repérer dans les différentes phases, mais aussi de comprendre l'utilisation de l'application sous deux éléments que nous allons vous expliquer ci-dessous.
- Mise en place d'un tutoriel et d'un objectif calorique : Pour cela, dans cet userform, vous pouvez disposer d'un tutoriel à votre guise afin de vous expliquer et de vous

procurer une aide pour naviguer entre les différentes étapes. Dans un deuxième temps il y a également la possibilité de vous définir un objectif calorique afin de vous permettre de suivre votre niveau de consommation plus aisément.

- Ajout de Message Box : En effet, cela a été rajouté pour prévenir l'utilisateur de son avancement et de lui informer de l'étape dans laquelle il se situe, lui procurant un sentiment d'être guidé et un confort plus optimal dans son utilisation.

6) Récapitulatif des étapes de code

L'objectif de ce projet est de pouvoir automatiser les différentes étapes et les différents éléments qui constituent ces étapes, à partir de ce que l'on appelle des variables.

Pour commencer cette application, il a été décidé de répartir l'ensemble des variables (qui dans ce projet représente un produit, comme par exemple une carotte) sous 3 grands userform nommés : « Entrée », « Plat principale », et « Dessert ». Ces 3 grands Userform sont similaires pour créer une base stable afin de faciliter la compréhension pour l'utilisateur.

Pour mieux comprendre notre argumentation, nous avons séparé nos idées sous 6 étapes clés qui se suivent successivement

Etape 1-1 : Coder la première variable

Pour ce faire, il a fallu commencer par définir les variables afin de pouvoir ensuite les utiliser dans un userform. Ainsi, nous avons travaillé sur un premier userform « Plat principal » et une première variable nommée « Gigot d'agneau »

Voici l'étape de codage ci-dessous pour le Gigot d'agneau

```
Private Sub Agneau_Click()
```

```
    ' Déclarer la variable publique pour stocker le nom de l'aliment
```

```
    ' Données les données pour chaque variables
```

```
    AlimentSelectionne = " Gigot d'agneau"
```

```
    ProteineIndique = 20
```

```
    LipidesIndique = 21
```

```
    GlucidesIndique = 0
```

```
    KcalIndique = 269
```

```
    ' Ouvrir le userform calcul
```

```
    Me.Hide
```


calcul.Show vbModeless

End Sub

Private Sub ImageAgneau_Click()

' Déclarer la variable publique pour stocker le nom de l'aliment

' Données les données pour chaque variables

AlimentSelectionne = " Gigot d'agneau"

ProteineIndique = 20

LipidesIndique = 21

GlucidesIndique = 0

KcalIndique = 269

' Ouvrir le userform calcul

Me.Hide

calcul.Show vbModeless

End Sub

Suite à cette étape, toutes les autres variables ont été réalisées sur cette même maquette, en changeant les valeurs nutritionnelles et aussi le nom de la variable.

De plus, ces variables ont été déclarées en public afin de permettre l'enregistrement des valeurs lorsque vous cliquez sur l'image ou le texte de la variable sur tous les userform.

Ainsi le codage de cette intervention correspond au code ci-dessous pour la déclaration de variable :

Public AlimentSelectionne As String

variable qualitatif

et les valeurs nutritionnelles qui sont sauvegardé avec :

Public ProteineIndique As Single

variable quantitatif

Public GlucidesIndique As Single

Public LipidesIndique As Single

Public KcalIndique As Single

As single : signifie une déclaration de variable quantitatif

As string : signifie une déclaration de variable qualitatif

Etape 1-2 : Le calcul de la variable

Ensuite, nous avons pour projet que lorsque l'on clique sur l'image ou le texte de l'aliment, cela ouvre un autre userform (calcul) spécifique à celle-ci afin de permettre d'ouvrir une feuille de calcul. En soit, pour gagner du temps et éviter tout oubli sur un userform, nous nous sommes basé sur un userform (calcul) pour tous les aliments.

De ce fait, l'automatisation du nom en tant que titre de la variable a du être choisi en fonction de l'aliment sélectionné. Ce code permet ainsi d'avoir le nom de l'aliment dans le label pour chaque variable cliquée.

Le code de ce point correspond à :

```
Private Sub UserForm_Activate()  
  
    MsgBox "Veuillez indiquer la quantité  :" & AlimentSelectionne  
    Label1.Caption = AlimentSelectionne  
  
End Sub
```

Après avoir automatisé le nom de la variable pour le titre de l'userform calcul , nous nous sommes penché sur le calcul pour chaque valeur nutritionnelle de chaque aliment en nous basant sur une base de 100g pour toutes les variables. Le calcul correspond tout bonnement à un produit en croix dont vous pouvez constater le code ci-dessous.

'produit en croix pour le calcul des valeurs nutritionnelles

```
ProteinesTotal = ProteineIndique * Quantite / 100  
GlucidesTotal = GlucidesIndique * Quantite / 100  
LipidesTotal = LipidesIndique * Quantite / 100  
KcalTotal = KcalIndique * Quantite / 100
```

Pour ce faire, les variables ont été déclarées en privé dans le userform afin d'éviter qu'elles ne sortent pas du userform en comparaison avec les autres variables qui peuvent être sélectionnées.

Déclaration des variables en privé :

```
Dim ProteinesTotal As Single  
Dim GlucidesTotal As Single  
Dim LipidesTotal As Single
```

Dim KcalTotal As Single

Par la suite de cette étape, il nous a fallu les afficher dans les différents labels de chaque variable, pour permettre à l'utilisateur de les visualiser (voir à la page suivante).

' Afficher les valeurs nutritionnelles dans le Label

```
RatioProt.Caption = ProteinesTotal & "g"  
RatioKcal.Caption = KcalTotal & "kcal"  
RatioLip.Caption = LipidesTotal & "g"  
RatioGlu.Caption = GlucidesTotal & "g"
```

Etape 2-1 : Les calculs pour toutes les variables

Une fois que les étapes précédentes ont été réalisées, nous avons dirigé les résultats du calcul dans le userform nommé « Résultat » qui aux prémices de l'application avait pour unique rôle d'être utilisé comme vérification pour le seul aliment codé. Ce n'est que par la suite que nous avons rajouté un deuxième aliment nommé « riz », pour pouvoir mettre en relation plusieurs variables.

Pour débiter cette étape nous avons du mettre en place un bouton enregistrement pour le ou les aliments dans le userform « calcul », de la façon suivante.

```
ProteinesTotalAliment = ProteinesTotal  
GlucidesTotalAliment = GlucidesTotal  
LipidesTotalAliment = LipidesTotal  
KcalTotalAliment = KcalTotal
```

Et par la suite additionner les différents aliments qui peuvent être sélectionnés par la création d'une boucle à partir de la somme des valeurs nutritionnelles de ou des variables.

Cette boucle s'affiche de la façon suivante :

' crée une boucle pour la sommes des valeurs nutritionnelles du repas

```
ProteinesTotalRepas = ProteinesTotalRepas + ProteinesTotalAliment  
GlucidesTotalRepas = GlucidesTotalRepas + GlucidesTotalAliment  
LipidesTotalRepas = LipidesTotalRepas + LipidesTotalAliment  
KcalTotalRepas = KcalTotalRepas + KcalTotalAliment
```

De ce fait, le calcul de la somme des valeurs nutritionnelles permet au bouton d'actualiser à chaque fois qu'il est pressé les résultats, et ensuite de les envoyer dans le userform « résultat » qui affiche les différentes valeurs.

Code d'affichage des valeurs du repas dans le userform résultat :

' Afficher les totaux des valeurs nutritionnelles du plat

TotalProtPlat.Caption = ProteinesTotalRepas & "g"

TotalGluPlat.Caption = GlucidesTotalRepas & "g"

TotalLipPlat.Caption = LipidesTotalRepas & "g"

TotalKcalPlat.Caption = KcalTotalRepas & "kcal"

Etape 2-2 : Uniformisation des calculs pour les 3 userforms

Une fois que ces premières étapes ont été réalisées, nous avons pu effectuer ce codage pour d'autres variables situées dans les userforms « Entrée » et « Dessert », en le faisant avec peu de choix comme les étapes précédentes sauf certaine déclaration de variable dans un premier temps pour pouvoir corriger les erreurs plus facilement.

Dans le userform résultat :

Le code pour afficher les variables des « Entrée » et « Dessert » dans les différents labels:

' Afficher les totaux des valeurs nutritionnelles de l'entree

TotalProtEntree.Caption = ProteinesTotalEntree & "g"

TotalGluEntree.Caption = GlucidesTotalEntree & "g"

TotalLipEntree.Caption = LipidesTotalEntree & "g"

TotalKcalEntree.Caption = KcalTotalEntree & "kcal"

' Afficher les totaux des valeurs nutritionnelles du dessert

TotalProtDessert.Caption = ProteinesTotalDessert & "g"

TotalGluDessert.Caption = GlucidesTotalDessert & "g"

TotalLipDessert.Caption = LipidesTotalDessert & "g"

TotalKcalDessert.Caption = KcalTotalDessert & "kcal"

Lorsque l'on a obtenu les différents totaux pour les variables sélectionnées, nous avons pu commencer à mettre en place la somme des totaux du repas afin d'avoir une lecture plus précise et agréable pour l'utilisateur et de les afficher dans différents labels. Voici le codage ci-dessous.

' Afficher les totaux des valeurs nutritionnelles du repas

TotalProt.Caption=(ProteinesTotalRepas + ProteinesTotalEntree + ProteinesTotalDessert) & "g"

TotalGlu.Caption = (GlucidesTotalRepas + GlucidesTotalEntree + GlucidesTotalDessert) & "g"

TotalLip.Caption = (LipidesTotalRepas + LipidesTotalEntree + LipidesTotalDessert) & "g"

```
TotalKcal.Caption = (KcalTotalRepas + KcalTotalEntree + KcalTotalDessert) & "kcal"
```

A partir de cette étape, la base même du projet est atteinte et permet de se concentrer sur d'autres aspects en parallèle et plus particulièrement de l'application.

Etape 3 : La gestion des erreurs

En effet, les utilisateurs peuvent ne pas comprendre un élément, et donc ils faut les guider en initiant des userforms particuliers avec ce qu'ils ont le droit ou non de faire. Pour cela nous avons commencé par gérer la gestion des erreurs dans le userform de « calcul ».

L'objectif de cette gestion, se caractérise par le fait que si l'utilisateur saisisait un point à la place de la virgule , une lettre à la place d'un chiffre ou un chiffre négatif, l'opération ne serait pas possible. Pour cette opération, vous pouvez voir ci-dessous le codage de cette gestion d'erreur.

' Remplacer les points par des virgules si présents

```
TextQuantite.Value = Replace(TextQuantite.Value, ".", ",")
```

' Gestionnaire d'erreur pour les nombres négatifs et les variables qualitatifs

```
If IsNumeric(TextQuantite.Value) Then
```

```
    Quantite = TextQuantite.Value
```

```
If Quantite < 0 Then
```

```
    MsgBox "La quantité ne peut pas être négative.", vbExclamation
```

```
    TextQuantite.SetFocus
```

```
    TextQuantite.Value = "" ' Réinitialise la valeur du champ de texte
```

```
    Exit Sub
```

```
End If
```

Pour la 1^{ère} ligne du code : cela modifie la valeur de la texte box si elle a un point et le mettre en virgule.

Pour les autres lignes sert pour si les valeurs sont négatifs ou qualitatifs .

Ainsi, si l'erreur survenait, une msgbox (message box) s'afficherait pour vous expliquer que l'opération de l'utilisateur n'était pas possible et donc de recommencer une saisie valide pour pouvoir continuer sa navigation dans l'application en le renvoyant sur la textbox dans laquelle il se situait et en supprimant ce qu'ils avaient mis.

Voici la suite du code pour la textbox :

```
Else
```

```
    MsgBox "Veuillez entrer une quantité valide.", vbExclamation
```

```
    TextQuantite.SetFocus
```

```
    TextQuantite.Value = "" ' Réinitialise la valeur du champ de texte
```

```
    Exit Sub
```

End If

De plus, si l'utilisateur ne met pas de données chiffrées dans la textbox et que celui-ci décide de cliquer pour essayer d'ajouter un autre aliment dans son repas, cette action engendrerait un message dit « erreur » pour le prévenir de son opération impossible à réaliser dans l'application.

Voici le codage de cette erreur (dont l'autre partie sur la page suivante) :
Nous l'avons mis dans le bouton d'enregistrement.

```
If TextQuantite.Value = "" Then  
    MsgBox "Vous n'avez pas mis de quantité.", vbExclamation  
    TextQuantite.SetFocus ' Dirige l'utilisateur vers le contrôle pour entrer la quantité
```

Pour conclure ce point, nous voulions faire en sorte que dans le userform appelé « Résultat », un label avec toutes les variables que l'utilisateur a sélectionné puisse s'afficher pour entretenir une clarté et une meilleure compréhension du repas qui a été choisi. Ce point particulier du projet a été plus complexe à réaliser dû à l'insertion d'une autre boucle, mais aussi d'une collection.

Pour ce faire, nous nous sommes dirigé vers le userform « calcul » afin d'obtenir les données brutes des aliments que l'utilisateur avait sélectionné et c'est à partir de ce moment que nous avons créé une collection pour différencier les aliments.

Pour ce faire nous avons créé une première sélection principale :

```
Public AlimentsSelectionnes As Collection
```

Et par conséquent, nous avons dû l'activer directement à partir de l'activation de l'activité, avec :

```
Set AlimentsSelectionnes = New Collection
```

Et la quantité a été déclarée en public pour pouvoir déplacer la variable dans les autres auserforms.

```
Public Quantite As Single
```

Etape 4 : Le transfert des données des aliments vers les collections

Une fois ces étapes conclues sur les différents modules, nous avons été sur le userform « Calcul » pour pouvoir enregistrer les différentes données que l'utilisateur a sélectionné dans une sous-rubrique de la collection afin de créer un meilleur aspect visuel. En effet, les données ont été récupérées des modules pour les ajouter à la collection principale. Le code est inséré dans le bouton d'enregistrement du calcul.

Pour cela voici le code pour enregistrer les valeurs dans la collection :

```
' Récupérer les informations sur l'aliment sélectionné
Dim Aliment As New Collection
Aliment.Add AlimentSelectionne, "Nom"
Aliment.Add Quantite, "Quantite"
Aliment.Add ProteinesTotalAliment, "Proteines"
Aliment.Add GlucidesTotalAliment, "Glucides"
Aliment.Add LipidesTotalAliment, "Lipides"
Aliment.Add KcalTotalAliment, "Kcal"

' Ajouter l'aliment à la collection principale AlimentsSelectionnes
AlimentsSelectionnes.Add Aliment
```

Maintenant que ces valeurs ont été ajoutées dans la collection, l'objectif final se caractérise par le fait d'afficher votre repas avec les valeurs nutritionnelles rattachées. De ce fait, nous allons donc les afficher dans le userform « Résultats » de la manière suivante :

```
' Effacer le contenu du label avant d'afficher les résultats
AlimentPlat.Caption = ""

' Utiliser une boucle For Each pour parcourir la collection AlimentsSelectionnes
For Each Aliment In AlimentsSelectionnes
    ' Vous pouvez accéder aux éléments de chaque aliment en utilisant Aliment
    ' Chaque aliment revient à la ligne avec toutes leurs variables
    AlimentPlat.Caption = AlimentPlat.Caption & "Aliment: " & Aliment("Nom") & " | " & _
        "Quantité: " & Aliment("Quantite") & "g | " & _
        "Kcal: " & Aliment("Kcal") & "kcal | " & _
        "Protéines: " & Aliment("Proteines") & "g | " & _
        "Lipides: " & Aliment("Lipides") & "g | " & _
        "Glucides: " & Aliment("Glucides") & "g" & vbCrLf & vbCrLf
Next Aliment
```

L'explication du code : les valeurs sont enregistrées dans une collection puis envoyées dans un label dans le userform résultat. Chaque aliment enregistré est mis dans le même label mais est espacé de 2 lignes.

Tout d'abord, dans le label « Plat principal » tous les plats principaux ou les aliments étaient déjà affichés ainsi que toutes les données nécessaires pour l'utilisateur. Néanmoins, il nous

manquait les « Entrée » et les « Dessert ». Pour cela nous avons commencé par les desserts mais nous nous sommes heurté à des problèmes. C'est ainsi, que la création d'une collection a été décidée pour les 3 grands userform qui sont « Entrée », Plat principal » et « Dessert », dont le codage est affiché ci-dessous.

```
Public AlimentsSelectionnes As Collection
Public AlimentsSelectionnesDessert As Collection
Public AlimentsSelectionnesEntree As Collection
```

Ainsi nous avons pu finaliser la conception et le rattachement des données pour le userform « Résultat ».

Etape 5 : Intégration du repas dans la page Excel

En effet, nous avons décidé de transmettre ce tableau de données depuis la VBA vers une page Excel en utilisant le même modèle que le userform « Résultat » afin d'avoir une lecture plus rapide pour le lecteur. Pour ce faire, nous avons créé ce codage ci-dessous représentant la déclaration des variables dans le bouton (enregistré mon repas) .

'déclaration variable intégration à une page Excel

```
Dim feuille As Worksheet
Dim ligne As Long
Dim lastRow As Long
Dim titre As Range
Dim rng As Range
```

Ensuite, il a fallu ouvrir une feuille repas s'il n'en n'existe déjà pas pour pouvoir y transférer les valeurs nutritionnelles avec ce codage ici présent :

' Création ou sélection de la feuille "Repas"

```
On Error Resume Next
Set feuille = ThisWorkbook.Sheets("Repas")
On Error GoTo 0

If feuille Is Nothing Then
    ' Si la feuille n'existe pas, créez-la
    Set feuille =
ThisWorkbook.Sheets.Add(After:=ThisWorkbook.Sheets(ThisWorkbook.Sheets.Count))
    feuille.Name = "Repas"

End If
```


Dans cette feuille de calcul nous avons également voulu y ajouter un titre afin de différencier et séparer les 2 tableaux. Celui-ci s'affiche automatiquement par un fond blanc et se place directement sur la dernière ligne et pour ce faire nous avons dû le réaliser de la façon qui est écrite ci-dessous.

With feuille

' Commencer l'insertion du repas à la dernière ligne

ligne = .Cells(.Rows.Count, "B").End(xlUp).Row

ligne = ligne + 2

Set titre = feuille.Range(feuille.Cells(ligne, 1), feuille.Cells(ligne, 7))

' Création du titre du repas avec la bordure et fonds blanc

.Range(.Cells(ligne, 1), .Cells(ligne, 7)).Merge

.Cells(ligne, 1).Value = " Mon Repas "

.Cells(ligne, 1).Font.Size = 16

.Cells(ligne, 1).Font.Bold = True

.Range(.Cells(ligne, 1), .Cells(ligne, 7)).Borders.LineStyle = xlContinuous

.Range(.Cells(ligne, 1), .Cells(ligne, 7)).Borders.Color = RGB(0, 0, 0)

La première ligne du code sert à trouver la dernière ligne utilisé et de pouvoir écrire 2 ligne après cette ligne .

Et pour une meilleure compréhension les colonnes ont également eu cette phase de titrisation avec des éléments en gras et surtout de placer les valeurs qui ont été transférées vers le document Excel de façon organisée. De plus, ce processus a été répété pour les userforms « Dessert » et « Entrée ».

Le codage de cette phase :

' Ajout des titres des colonnes pour l'entrée

With feuille

' titre entrée en gras

feuille.Cells(ligne, 1) = "Entrée"

.Cells(ligne, 1).Font.Bold = True

' Titres des colonnes

.Cells(ligne + 1, 2).Value = "Aliment"

.Cells(ligne + 1, 3).Value = "Quantité (g)"

.Cells(ligne + 1, 4).Value = "Kcal"

.Cells(ligne + 1, 5).Value = "Protéines (g)"

.Cells(ligne + 1, 6).Value = "Glucides (g)"

.Cells(ligne + 1, 7).Value = "Lipides (g)"

ligne = ligne + 2

End With

Puis nous avons insérer chaque valeurs de chaque aliments dans les colonnes correspondantes et mettre un chiffre après la virgule. Chaque aliment à sa propre ligne .

' Exportation des données de l'entrée dans la feuille "Repas"

For Each AlimentEntree In AlimentsSelectionneesEntree

With feuille

.Cells(ligne, 2).Value = AlimentEntree("Nom")

.Cells(ligne, 3).Value = AlimentEntree("Quantite")

.Cells(ligne, 4).Value = AlimentEntree("Kcal")

.Cells(ligne, 5).Value = AlimentEntree("Proteines")

.Cells(ligne, 6).Value = AlimentEntree("Glucides")

.Cells(ligne, 7).Value = AlimentEntree("Lipides")

' Formatage spécifique pour les valeurs Kcal, Lipides, Protéines et Glucides

.Cells(ligne, 4).NumberFormat = "0.0" ' Un chiffre après la virgule

.Cells(ligne, 5).NumberFormat = "0.0" ' Un chiffre après la virgule

.Cells(ligne, 6).NumberFormat = "0.0" ' Un chiffre après la virgule

.Cells(ligne, 7).NumberFormat = "0.0" ' Un chiffre après la virgule

ligne = ligne + 1

End With

'passer à un autre aliment de l'entrée

Next AlimentEntree

Nous avons réalisé les 2 dernière étapes pour le plat principal et le dessert.

Enfin, nous avons inséré une ligne totale pour les repas pour pouvoir suivre les résultats provenant du userform « Résultat », que vous pouvez voir ci-dessous.

With feuille

.Cells(ligne, 1).Value = "Repas "

.Cells(ligne, 1).Font.Bold = True

.Cells(ligne + 1, 2).Value = "Totaux"

.Cells(ligne + 1, 4).Value = TotalKcal.Caption

.Cells(ligne + 1, 5).Value = TotalProt.Caption

.Cells(ligne + 1, 6).Value = TotalGlu.Caption

.Cells(ligne + 1, 7).Value = TotalLip.Caption

End With

Une fois cette étape conclue, il a fallu mettre en place un code qui évite que les repas ne se chevauchent, c'est ainsi que nous avons mis en place un fond gris et appliqué d'autres étapes

de mise en forme comme le centrage ou encore définir la colorisation des cellules des titres en blanc. Cette étape est visible ci-dessous.

```
' Mettre en gris les cellules de A à G pour le repas
lastRow = feuille.Cells(feuille.Rows.Count, "B").End(xlUp).Row
Set rng = feuille.Range(feuille.Cells(4, 1), feuille.Cells(lastRow, 7))
' Mise en forme du fond gris pour les données
With rng.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .Color = RGB(220, 220, 220) ' Couleur de remplissage gris clair
End With

' Changer la couleur de fond des cellules spécifiques pour le titre, les mettant en blanc
titre.Interior.Color = RGB(255, 255, 255)

' Centrer toutes les cellules
feuille.Cells.HorizontalAlignment = xlCenter

' Ajuster la largeur des colonnes en fonction du contenu
feuille.Columns("A:G").AutoFit
```

En finalisant ces étapes, cela permet de conclure le raccord de la transition des données du codage de la VBA vers le lien Excel.

Etape 6 : La mise en forme de l'application

Pour pouvoir présenter et utiliser cette application nous avons fait le choix de soigner la mise en forme. En effet cet aspect peut paraître optionnel, néanmoins pour la clarté et la compréhension de l'application il est important.

Pour commencer cette étape, nous avons vérifié la mise en place du bon fonctionnement de chaque aliment, mais également l'intégration d'un bouton dit « Accueil » et d'un objectif calorique pour le repas non obligatoire. Ces éléments constituent la première page qui est utile afin de laisser le temps à l'utilisateur de se projeter dans l'application.

Concernant l'objectif calorique, nous avons tout d'abord déclaré la variable de façon suivante : « Public Objectif As Single », qui permet d'être relié au userform « Résultat ».

Ainsi nous avons affiché un textbox pour que l'utilisateur puisse entrer ces choix d'objectifs, qui a été codé de la façon suivante :

```
TextObjectif.Caption = Objectif & "kcal"
```

Et enfin l'affichage du code qui permet de savoir si l'individu a tenu ses objectifs qu'il c'était fixé, et de ce fait, pouvoir les changer si besoin.

```
If Objectif < KcalTotalRepas + KcalTotalEntree + KcalTotalDessert Then  
TextFinalObjectif.Caption = "Surplus Calorique"
```

```
ElseIf Objectif > KcalTotalRepas + KcalTotalEntree + KcalTotalDessert Then  
TextFinalObjectif.Caption = "Déficit Calorique"
```

```
ElseIf Objectif = KcalTotalRepas + KcalTotalEntree + KcalTotalDessert Then  
TextFinalObjectif.Caption = "Objectif atteint"
```

```
End If
```

Pour finir l'application, nous avons rajouté dans « Accueil » un tutoriel pour laisser le choix à l'utilisateur de pouvoir prendre le temps avec une explication plus détaillée de l'application avant de commencer à l'utiliser .

Enfin dans le userform résultat nous avons rajouté 2 message box pour les 2 boutons , soit retour et soit enregistré mon repas qui permet à l'utilisateur de quitté l'application soit en ayant enregistré le repas soit sans enregistrer.

Pour le bouton annuler :

```
If MsgBox("Voulez vous quitter l'application?", vbYesNo) = vbYes Then  
Me.Hide
```

```
    ' Masquer le UserForm3
```

```
ElseIf vbNo Then
```

```
    ' Afficher à nouveau le UserForm1
```

```
    Me.Hide
```

```
    Dessert.Show
```

```
End If
```

```
End Sub
```

On a fait une message box à 2 sortie soit il revient sur la page dessert , soit il quitte l'application.

Dans le bouton enregistré mon repas :

Nous avons placé ce code pour enregistrer le repas si il clique sur quitter l'application avec un oui .

```
If MsgBox("Vous avez bien mémorisé votre repas !" & vbNewLine & vbNewLine & " Voulez-vous quitter l'application?", vbYesNo) = vbYes Then  
Me.Hide
```

Suite du code...

```
End if
```

Si la personne clique sur non cela retourne sur le userform résultat.

7) Conclusion

En résumé, le cahier des charges pour "Nutrition Bourse" a été établi pour définir les spécifications techniques et fonctionnelles du projet. Il décrit l'application en sept étapes, avec la flexibilité nécessaire pour passer d'une étape à l'autre sans avoir à choisir d'aliments à chaque étape. L'objectif ultime est de fournir un outil convivial pour aider les utilisateurs à composer des repas équilibrés et à comprendre la nutrition.

D'une manière moins formelle, cette application est conçue pour être conviviale et flexible. Vous pouvez passer d'une étape à l'autre à votre rythme, sans obligation de choisir des aliments à chaque étape. Vous pouvez également sélectionner autant d'aliments que vous le souhaitez. De ce fait, l'objectif final est d'aider à mieux comprendre la nutrition et à prendre des décisions éclairées sur les repas afin de satisfaire les différentes attentes des utilisateurs.

Maintenant de notre point de vue ce projet nous a apporté la compréhension et l'utilisation de l'outil de code sur Excel VBA . Une meilleur logique et une résilience fasse à l'incompréhension des problèmes de codage que si n ne peut pas passer par la porte parce qu'on ne sait pas faire alors on passera par une fenêtre jusqu'à réussir.

Merci Mme Pain de nous avoir permis de rajouter une corde à notre arc .