



RAPPORT PROJET

2019 - 2020

Application d'inscription à un concours national

Christian Gamo 102 & Monica Huynh 102

Encadrés par M. Rossit et M. Alès-Bianchetti

TABLE DES MATIÈRES

INTRODUCTION

APPLICATION

Utilisation de l'application.....	4-15
Schéma d'ordonnancement.....	16
Exemple des fichiers bilans.....	17-18
Code source des formulaires.....	19-43

CONCLUSION

INTRODUCTION

Présentation de l'application

L'objectif de l'application est de recueillir toutes les inscriptions à un concours national et d'en faire un bilan.

Les fonctionnalités

1. Enregistrer une inscription

Permet d'enregistrer un nouveau candidat. Trois étapes : saisie des renseignements du candidat, les choix des différentes matières (à l'écrit et à l'oral) et de la région et enfin, la confirmation après un récapitulatif global.

2. Modifier ou supprimer un candidat

Permet de modifier ou de supprimer un candidat existant. On accède aux informations du candidat soit par son unique numéro ou par son nom et prénom.

3. Afficher l'état actuel des inscriptions

Affiche la liste des candidats inscrits triée par ordre alphabétique des noms puis des prénoms.

4. Consulter le bilan des inscriptions

Permet de consulter le bilan pour une matière ou pour une inscription. Il peut être *final* si les inscriptions sont terminées ou *provisoire* si ce n'est pas le cas.

Pour rappel, notre concours se déroule en France et concerne les 10 régions suivantes : *Auvergne, Bordelais, Bourgogne, Bretagne, Corse, Nord, Normandie, Paris, Poitou et Roussillon*. Les candidats participent à ce concours spécialisé sur le thème de l'Informatique mais qui veille à connaître les connaissances des élèves sur d'autres matières plus générales. Voici la liste des matières proposées : Algorithmique, Base de données, **Droit**, **Expression**, **Gestion**, Langage C, Langage Java, **Mathématiques**, Programmation web, **Réseau**, **Système**, Visual Basic .NET, **Anglais**, **Chinois** et **Espagnol**. *En noir, écrit - en **jaune**, écrit & oral - en **orange**, oral.*

APPLICATION

Après une introduction présentant le contexte de l'application ainsi que ses fonctionnalités, on détaille maintenant le « mode d'emploi » de l'application. Nous veillons à une bonne compréhension de l'utilisation des différentes fonctionnalités.

Utilisation de l'application

Accueil (Accueil.vb)

Au démarrage de l'application, l'utilisateur tombe sur le menu général : « l'Accueil » où il peut accéder aux différentes fonctionnalités présentées précédemment par un simple clic sur l'un des boutons. En plus des fonctionnalités principales, nous pouvons choisir de « Quitter » l'application (le dernier bouton) en plein milieu des inscriptions. C'est-à-dire que l'utilisateur peut se permettre de sortir de l'application par ce bouton même s'il n'a pas fini d'inscrire les candidats, une sauvegarde s'effectuera automatiquement lui permettant de reprendre son travail là où il en était. La sauvegarde est constituée de 2 fichiers « sauvegarde.bin » et « nbCpt.bin ». Le premier donne la liste des candidats et le deuxième donne le compteur du numéro attribué à l'inscription. Ils seront situés dans le dossier bin/Debug du projet.



Figure 1. Menu général

Enregistrer une inscription : Renseignements (InfosCandidat.vb)

Sur l'accueil, si on clique sur « Enregistrer une inscription », une fenêtre s'ouvre proposant de saisir les informations d'un nouveau candidat et un compte à rebours de 60 secondes est lancé. L'heure et le temps restant sont indiqués dans la barre de titre.

Les champs obligatoires sont indiqués par un astérisque (*). Si une information est incomplète ou erronée, le label de celle-ci deviendra rouge pour signaler l'oubli.

L'âge est sélectionné grâce à une barre de défilement horizontale.

Il y a également deux boutons : « Abandon » et « Suite » qui permettent, respectivement, d'abandonner l'inscription et de passer à la deuxième étape de l'inscription.

Figure 2. Renseignements

Une fois le compte à rebours atteint, on retourne à l'écran d'accueil après l'apparition d'un message.

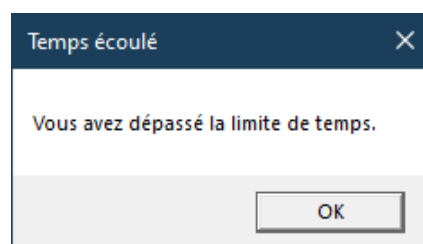


Figure 3. Temps écoulé

Enregistrer une inscription : Choix des épreuves (ChoixEpreuves.vb)

En cliquant sur « Suite », l'utilisateur doit sélectionner les matières des différentes épreuves que le candidat a choisi. Se présente une liste déroulante qui contient les 10 régions précisées dans l'introduction, le nom et le prénom du candidat sont également affichés. Les matières pour les épreuves écrites et orales sont organisées dans deux groupes différents, des cases à cocher permettent la sélection des matières. Il faut obligatoirement sélectionner 4 matières dans le groupe Écrit et 3 matières dans le groupe Oral, une fois cela fait, une proposition d'épreuve facultative (orale) apparaîtra. Il y a également un compte à rebours de 90 secondes qui possède les mêmes fonctionnalités que le précédent. Si on abandonne l'inscription, on retourne à **Accueil**.

Figure 4. Choix des épreuves

Figure 5. Choix de l'épreuve facultative

Précisions : Si on sélectionne une matière écrite qui est également une épreuve qui peut être passée à l'orale, on empêche la sélection de cette matière à l'écrit. Inversement, une épreuve orale ne peut être passée à l'écrit si elle a été choisie. Les épreuves facultatives sont les matières possibles à l'oral qui n'ont pas été sélectionnées ni à l'écrit, ni à l'oral.

Enregistrer une inscription : Récapitulatif (Recap.vb)

En cliquant sur « Enregistrer », une fenêtre récapitulative des informations du candidat et des épreuves choisies apparaît.

Si aucune adresse ou aucune épreuve orale facultative n'a été saisie, leurs zones apparaîtront vides et grisées.

Il y a également trois boutons : « Abandon », « Retour » et « Inscrire » qui permettent, respectivement d'abandonner l'inscription, de retourner au formulaire du choix des épreuves et de confirmer l'inscription du candidat. Dans ce dernier cas, un numéro unique est attribué au nouveau candidat. Si l'on souhaite retourner aux choix des épreuves, le compte à rebours est réinitialisé (à 90 secondes) et redémarre.

The screenshot shows a window titled "Récapitulatif" with a dark blue header. The main area has a light gray background and displays the following information:

- Nom:** Térieur
- Prénom:** Alain
- Région:** Auvergne
- Épreuve orale facultative:** Droit
- Code postal:** 93200
- Ville:** Saint Denis
- Épreuves écrites:** Algorithmique, Base de données, Langage C, Réseau
- Épreuves orales:** Expression, Gestion, Anglais
- Âge:** 23
- Adresse:** (empty field)

At the bottom, there are three buttons: "Abandon" (red), "Retour" (white), and "Inscrire" (green).

Figure 6. Récapitulatif de l'inscription d'un candidat

The screenshot shows a small dialog box titled "Attribution du numéro candidat" with a dark blue header and a close button (X). The main area is white and contains the text:

Le numéro qui vous a été attribué est le numéro : 1

At the bottom, there are two buttons: "OK" and "Annuler".

Figure 7. Attribution du numéro de candidat

Modification ou suppression d'un candidat (ModifSuppr.vb)

Sur l'accueil, si on clique sur « Modification ou suppression » on arrive sur cette fenêtre.



Figure 8. Modification ou suppression

On peut choisir de sélectionner un candidat soit par son numéro, soit par une sélection de son identifiant (numéro, nom, prénom) dans une liste déroulante.



Figure 9. Identifiant

Si l'utilisateur clique sur le bouton « Supprimer » ou le bouton « Modifier » alors qu'aucune information n'a été mise dans les champs, cette zone se grisera pour signaler un oubli. Au moment où l'utilisateur entrera une information, la zone redeviendra blanche. Le bouton « Retour » permet de revenir à l'écran d'accueil.

Suppression d'un candidat (Recap.vb)

En choisissant de supprimer un candidat, on ouvre la fenêtre **Récapitulatif** qui a été adaptée pour une suppression de candidat. On se repose sur la même base (un récapitulatif des informations du candidat). Deux boutons s'offrent à l'utilisateur : « Abandon » qui annule la suppression et « Supprimer » qui supprime le candidat, dans les deux cas nous revenons à la fenêtre **Modification ou suppression** d'un candidat.

The screenshot shows a Windows application window titled "Récapitulatif". The main content area has a light gray background with the word "Suppression" in a large, italicized, black serif font at the top center. Below the title, there are several input fields and text boxes arranged in a grid-like fashion. The fields are labeled as follows:

- Nom:** A text box containing "Téreur".
- Prénom:** A text box containing "Alex".
- Région:** A text box containing "Auvergne".
- Épreuve orale facultative:** An empty text box.
- Code postal:** A text box containing "93200".
- Ville:** A text box containing "Saint Denis".
- Épreuves écrites:** A text box containing the text "Gestion", "Langage C", "Système", and "Visual Basic .NET" stacked vertically.
- Épreuves orales:** A text box containing the text "Mathématiques", "Réseau", and "Espagnol" stacked vertically.
- Âge:** A text box containing "23".
- Adresse:** An empty text box.

At the bottom of the window, there are two buttons:

- A red button labeled "Abandon" on the left.
- A green button labeled "Supprimer" on the right.

Figure 10. Suppression d'un candidat

Modification des données d'un candidat (ChoixEpreuves.vb & Recap.vb)

En choisissant de modifier un candidat, on ouvre la fenêtre **Choix des épreuves** qui a été adaptée pour une modification de données : si on abandonne ici on retourne sur la fenêtre **Modification ou suppression**.

Figure 11. Choix des épreuves pour une modification des données

On enregistre ensuite les modifications, suivant le même principe que pour l'inscription, la fenêtre **Récapitulatif** s'ouvre.

Figure 12. Modification des données d'un candidat

Les boutons « Abandon » et « Modifier » mènent à **Modification ou suppression** et le bouton « Retour » à la figure 11.

Afficher l'état actuel des inscriptions (AffInscrip.vb)

Sur l'accueil, si on clique sur « Afficher l'état actuel des inscriptions », une fenêtre affichant la liste des candidats s'ouvre.

Cette liste est triée de manière alphabétique, d'abord des noms puis des prénoms, et apparaît sous la forme de 6 ListBox synchronisées entre elles, présentant successivement le numéro, le nom, le prénom, les épreuves écrites choisies, les épreuves écrites choisies et l'épreuve facultative si le candidat en a choisi une. Elles sont synchronisées grâce à une sélection synchronisé au clic et une unique barre de défilement verticale située tout à droite qui contrôle le défilement de chaque ListBox.

Les matières sont désignées par les 3 premières lettres de leur intitulé complet mis à part pour les matières Langage C et Langage Java pour pouvoir les différencier.

Un bouton « Retour » permet de retourner à l'écran d'accueil.

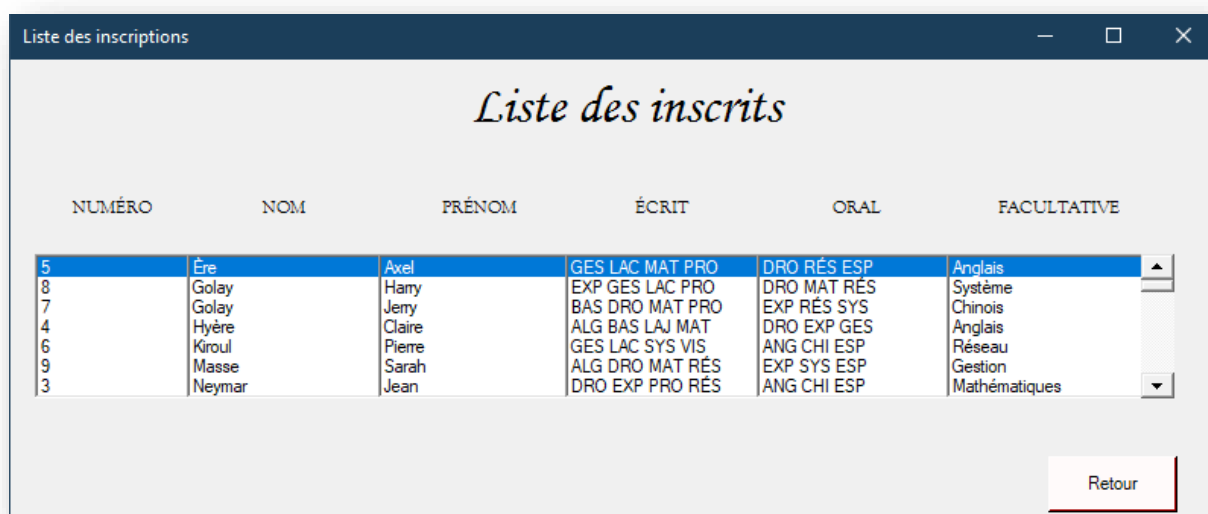


Figure 13. Liste des inscrits

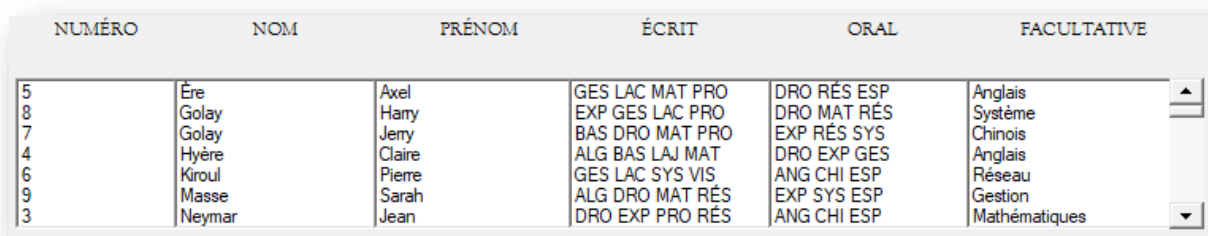


Figure 14. Synchronisation des ListBox

Bilan des inscriptions (Bilan.vb)

Sur l'accueil, si on clique sur « Bilan des inscriptions », une boîte de dialogue (MsgBox) s'ouvre demandant si l'on a terminé les inscriptions.

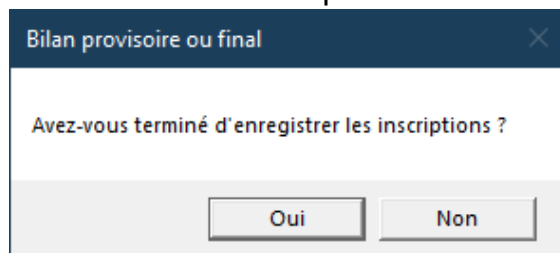


Figure 15. Boîte de dialogue

Une fenêtre permettant le choix du bilan s'ouvre. Si le bilan est provisoire, un bouton « Retour » permettant de retourner à l'accueil est proposé.



Figure 16. Choix du bilan provisoire

Si le bilan est au contraire final, un bouton « Quitter » est proposé. Si l'on clique sur ce bouton, le programme génèrera un ensemble de fichiers texte correspondant aux listings par épreuve des candidats inscrits à l'examen et quittera l'application.



Figure 17. Choix du bilan final

Si l'on coche la case « Individuel » on pourra choisir le candidat que l'on veut dans une liste déroulante.



Figure 18. Sélection du candidat

Si au contraire la case « Matière », on pourra choisir la matière que l'on veut dans une liste déroulante.

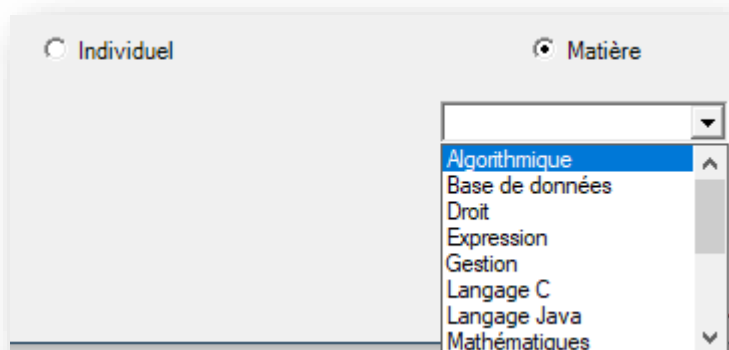


Figure 19. Sélection du candidat

Il est important de noter que lorsque l'on quitte le choix du bilan final, s'il y a une sauvegarde, elle sera supprimée.

Bilan individuel (Recap.vb)

Si on choisit le bilan d'un candidat, le formulaire **Récapitulatif** qui été adapté pour être un bilan individuel s'ouvre.

Un bouton « Retour » permet de retourner au choix du bilan.

The screenshot shows a Windows application window titled "Récapitulatif". Inside the window, the title "Bilan individuel" is displayed in a large, italicized serif font. Below the title, the form is organized into several sections with labels and input fields:

- Nom:** Térieur
- Prénom:** Alain
- Région:** Paris
- Épreuve orale facultative:** Droit
- Code postal:** 93200
- Ville:** Saint Denis
- Épreuves écrites:** Algorithmique, Base de données, Langage C, Réseau
- Épreuves orales:** Expression, Gestion, Anglais
- Âge:** 23
- Adresse:** (empty field)

In the bottom right corner of the form, there is a button labeled "Retour".

Figure 20. Bilan du candidat Alain Térieur

Bilan des matières (BilanMatiere.vb)

Si on choisit le bilan d'une matière, la fenêtre ci-dessous apparaît.

The screenshot shows a window titled "Bilan matière" with a dark blue header. Below the header, the title "Bilan matière" is displayed in a large, italicized font. A text box at the top contains the word "Droit". Below this, there are two sections: "ÉCRIT" and "ORAL". Each section contains three columns: "RÉGION", "NOM", and "PRÉNOM". The "ÉCRIT" section lists candidates: Bourgogne, Golay, Jerry; Bourgogne, Masse, Sarah; and Paris, Neymar, Jean. The "ORAL" section lists candidates: Auvergne, Ère, Axel; Auvergne, Hyère, Claire; Bourgogne, Golay, Harry; and Paris, Térieur, Alain. A "Retour" button is located at the bottom right.

	RÉGION	NOM	PRÉNOM
ÉCRIT	Bourgogne	Golay	Jerry
	Bourgogne	Masse	Sarah
	Paris	Neymar	Jean
ORAL	Auvergne	Ère	Axel
	Auvergne	Hyère	Claire
	Bourgogne	Golay	Harry
	Paris	Térieur	Alain

Figure 21. Bilan de la matière Droit

On aperçoit le nom de la matière sélectionnée dans la zone de saisie en haut. Dans cette fenêtre, on sépare les candidats inscrits dans la matière pour l'épreuve écrite et pour l'épreuve orale. On a ainsi 3 ListBox pour chacune des épreuves pour y mettre la région le nom et le prénom d'un candidat. Elles sont également synchronisées entre elles, par la barre de défilement et lors du clic. La liste est triée alphabétiquement par la région d'abord, puis par le nom et enfin le prénom. Le bouton « Retour » permet de revenir sur la fenêtre **Bilan** afin de consulter un nouveau bilan ou alors quitter définitivement l'application (ou revenir à l'accueil si les inscriptions ne sont pas terminées).

Schéma d'ordonnancement

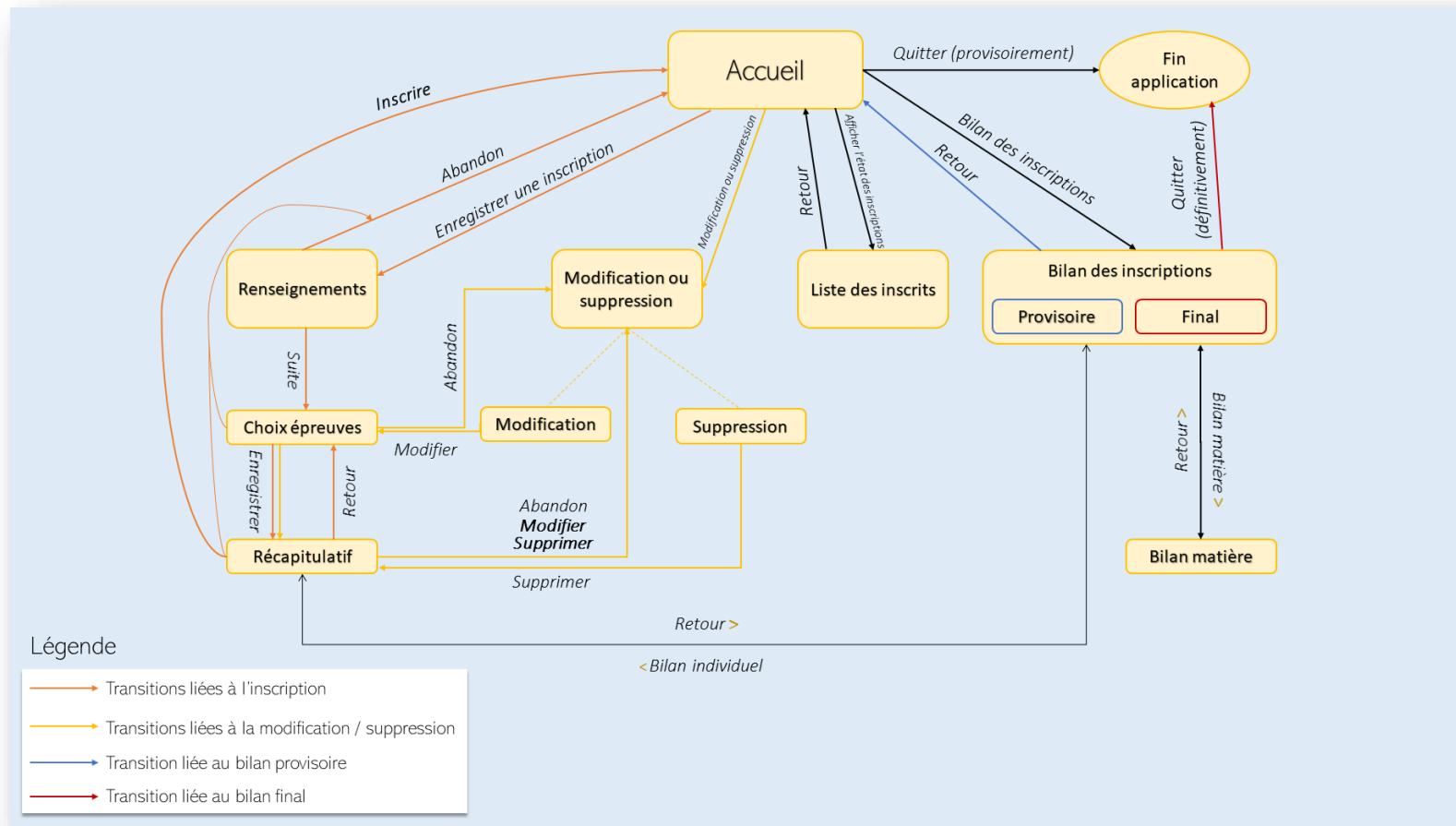
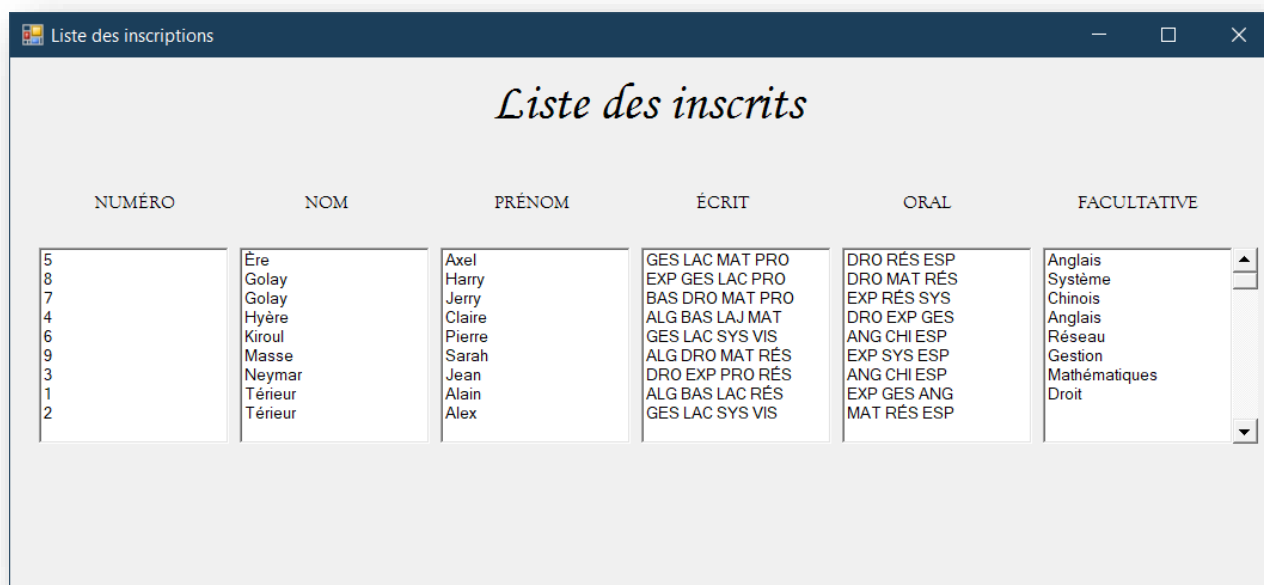


Figure 22. Schéma d'ordonnancement du projet

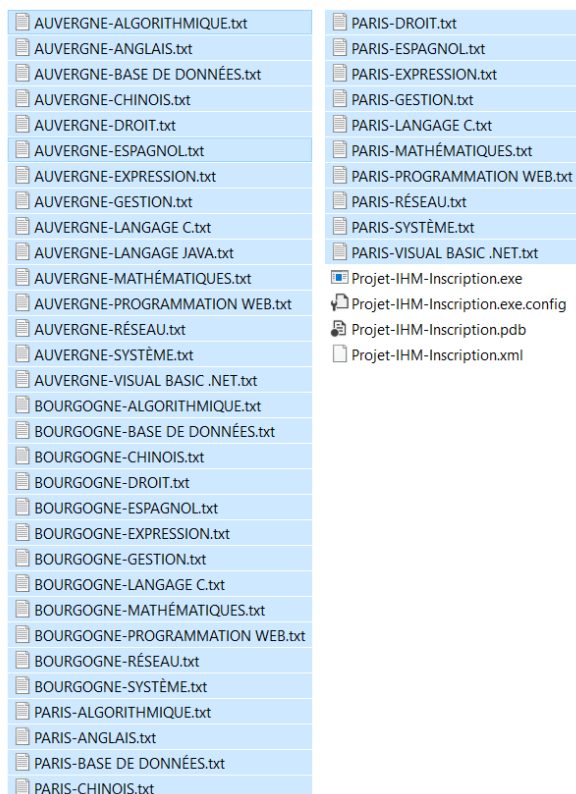
Un exemple des inscriptions enregistrées dans les fichiers bilans

Voici un exemple de 9 candidats au concours national avec leurs choix d'épreuves. Les candidats 1 à 3 sont de la région Paris, les candidats 4 à 6 sont de la région Auvergne et les candidats 7 à 9 sont de la région Bourgogne.



NUMÉRO	NOM	PRÉNOM	ÉCRIT	ORAL	FACULTATIVE
5	Ère	Axel	GES LAC MAT PRO	DRO RÉS ESP	Anglais
8	Golay	Harry	EXP GES LAC PRO	DRO MAT RÉS	Système
7	Golay	Jerry	BAS DRO MAT PRO	EXP RÉS SYS	Chinois
4	Hyère	Claire	ALG BAS LAJ MAT	DRO EXP GES	Anglais
6	Kiroul	Pierre	GES LAC SYS VIS	ANG CHI ESP	Réseau
9	Masse	Sarah	ALG DRO MAT RÉS	EXP SYS ESP	Gestion
3	Neymar	Jean	DRO EXP PRO RÉS	ANG CHI ESP	Mathématiques
1	Térier	Alain	ALG BAS LAC RÉS	EXP GES ANG	Droit
2	Térier	Alex	GES LAC SYS VIS	MAT RÉS ESP	

Figure 23. Liste des inscrits



AUVERGNE-ALGORITHMIQUE.txt	PARIS-DROIT.txt
AUVERGNE-ANGLAIS.txt	PARIS-ESPAGNOL.txt
AUVERGNE-BASE DE DONNÉES.txt	PARIS-EXPRESSION.txt
AUVERGNE-CHINOIS.txt	PARIS-GESTION.txt
AUVERGNE-DROIT.txt	PARIS-LANGAGE C.txt
AUVERGNE-ESPAGNOL.txt	PARIS-MATHÉMATIQUES.txt
AUVERGNE-EXPRESSION.txt	PARIS-PROGRAMMATION WEB.txt
AUVERGNE-GESTION.txt	PARIS-RÉSEAU.txt
AUVERGNE-LANGAGE C.txt	PARIS-SYSTÈME.txt
AUVERGNE-LANGAGE JAVA.txt	PARIS-VISUAL BASIC .NET.txt
AUVERGNE-MATHÉMATIQUES.txt	Projet-IHM-Inscription.exe
AUVERGNE-PROGRAMMATION WEB.txt	Projet-IHM-Inscription.exe.config
AUVERGNE-RÉSEAU.txt	Projet-IHM-Inscription.pdb
AUVERGNE-SYSTÈME.txt	Projet-IHM-Inscription.xml
AUVERGNE-VISUAL BASIC .NET.txt	
BOURGOGNE-ALGORITHMIQUE.txt	
BOURGOGNE-BASE DE DONNÉES.txt	
BOURGOGNE-CHINOIS.txt	
BOURGOGNE-DROIT.txt	
BOURGOGNE-ESPAGNOL.txt	
BOURGOGNE-EXPRESSION.txt	
BOURGOGNE-GESTION.txt	
BOURGOGNE-LANGAGE C.txt	
BOURGOGNE-MATHÉMATIQUES.txt	
BOURGOGNE-PROGRAMMATION WEB.txt	
BOURGOGNE-RÉSEAU.txt	
BOURGOGNE-SYSTÈME.txt	
PARIS-ALGORITHMIQUE.txt	
PARIS-ANGLAIS.txt	
PARIS-BASE DE DONNÉES.txt	
PARIS-CHINOIS.txt	

Voici les fichiers bilans enregistrés dans le dossier bin/Debug du projet, après avoir quitté le choix du bilan final.

Chaque fichier a pour nom l'intitulé de la matière ainsi que la région concernée.

Les fichiers correspondants aux épreuves pour lesquelles aucun candidat n'est inscrit ne sont pas créés.

Le titre de chaque fichier est du format « **PAYS - MATIÈRE** ».

Figure 24. Contenu du dossier bin/Debug

Ci-dessous se trouve un exemple d'un des fichiers bilan. Comme vous pouvez le voir, le contenu est divisé en 2 parties : la liste des candidats à l'écrit et la liste des candidats à l'oral.

Chaque section commence par un titre de la forme :

« Liste des candidats de la région **Région** pour la matière **Matière** à l'oral / l'écrit »

Le format de l'affichage des inscrits est de la forme : « Nom Prénom Numéro ».

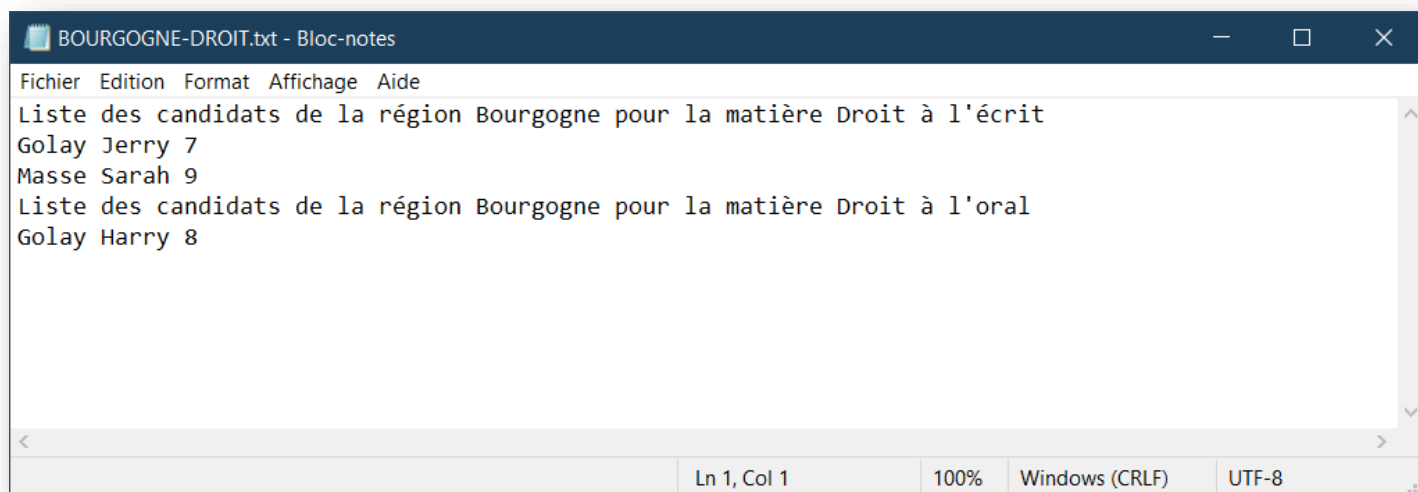


Figure 25. Contenu du fichier bilan BOURGOGNE - DROIT fin d'inscription

Si l'on redémarre l'application après la création des fichiers bilans, elles seront automatiquement supprimées pour laisser place à un nouveau concours national et donc de nouvelles inscriptions.

Codes sources des formulaires

- **ModuleCandidats.vb**

```
Imports System.Runtime.Serialization.Formatters.Binary
Imports System.IO

Module ModuleCandidats
    Private listeCandidats As List(Of candidat)
    Private cptInscrit As Integer = 0

    <Serializable()>
    Public Structure candidat
        Dim num As Integer
        Dim nom As String
        Dim prenom As String
        Dim ad As String
        Dim ville As String
        Dim cp As Integer
        Dim age As Integer
        Dim region As String
        Dim tabEpreuvesEcrit() As String
        Dim tabEpreuvesOral() As String
        Dim epFac As String
    End Structure
    'Getter pour la listeCandidats
    Public Function getListeCandidats()
        Dim copie As List(Of candidat)
        copie = listeCandidats
        Return copie
    End Function
    Public Function getCpt()
        Return cptInscrit
    End Function

    '@brief : Initialise la liste des candidats et lance l'application
    Sub Main()
        listeCandidats = New List(Of candidat)
        Dim nom As String = "sauvegarde.bin"
        Dim nomCpt As String = "nbCpt.bin"
        charger(nom, nomCpt)
        Application.Run(Accueil)
    End Sub
    '@brief : Ajoute un candidat à la liste des inscrits
    Public Sub Ajouter(ByRef candid As candidat)
        candid.num = cptInscrit + 1
        listeCandidats.Add(candid)
        MsgBox("Le numéro qui vous a été attribué est le numéro : " & candid.num, vbOK,
            "Attribution du numéro candidat")
        cptInscrit += 1
    End Sub
    '@brief : Modifie les informations d'un candidat
    Public Sub Modifier(ByRef candid As candidat, ByRef index As Integer)
        candid.num = listeCandidats(index).num
        listeCandidats(index) = candid
    End Sub
End Module
```

```

End Sub
'@brief : Supprime un candidat de la liste des inscrits
Public Sub Supprimer(ByRef index As Integer)
    listeCandidats.RemoveAt(index)
End Sub
'@brief : Créé une sauvegarde de la liste des candidats et du compteur de
candidats
Public Sub Sauvegarder(nomListe As String, nomCpt As String)
    Dim sauvegardeListe As FileStream = File.Create(nomListe)
    Dim sauvegardeCpt As FileStream = File.Create(nomCpt)
    Dim SerialiseurListe As New BinaryFormatter
    Dim SerialiseurCpt As New BinaryFormatter

    SerialiseurListe.Serialize(sauvegardeListe, listeCandidats)
    SerialiseurCpt.Serialize(sauvegardeCpt, cptInscrit)
    sauvegardeListe.Close()
    sauvegardeCpt.Close()
End Sub
'@brief: Récupère la sauvegarde de la liste et du compteur
Public Sub Charger(nomListe As String, nomCpt As String)
    If File.Exists(nomListe) And File.Exists(nomCpt) Then

        Dim FluxListe As Stream = File.OpenRead(nomListe)
        Dim DeserialiseurListe As New BinaryFormatter()
        Dim FluxCpt As Stream = File.OpenRead(nomCpt)
        Dim DeserialiseurCpt As New BinaryFormatter()

        listeCandidats = CType(DeserialiseurListe.Deserialize(FluxListe), List(Of
        candidat))
        cptInscrit = CType(DeserialiseurCpt.Deserialize(FluxCpt), Integer)

        FluxListe.Close()
        FluxCpt.Close()
    End If
End Sub

End Module

```

Accueil.vb

```
Imports System.IO
Public Class Accueil
    '@brief : Supprime les bilans liés à la fin des inscriptions d'un ancien concours
    Private Sub Accueil_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        If Not File.Exists("sauvegarde.bin") And Not File.Exists("nbCpt.bin") Then
            For Each fichierTxt In Directory.GetFiles(Directory.GetCurrentDirectory,
                "*.txt", SearchOption.TopDirectoryOnly)
                File.Delete(fichierTxt)
            Next
        End If
    End Sub
    '@brief : Sélectionne le menu Enregistrer, ouvre le formulaire pour inscrire un
    candidat
    Private Sub ButEnreg_Click(sender As Object, e As EventArgs) Handles
        ButEnreg.Click
            Me.Visible = False
            InfosCandidat.Visible = True
            Recap.SetEtat(Recap.EtatEnum.Enregistrer)
        End Sub
    '@brief : Ouvre le formulaire pour modifier/supprimer un candidat (s'il y en a)
    Private Sub ButtonModifSuppr_Click(sender As Object, e As EventArgs) Handles
        ButModifSuppr.Click
            Dim tab As List(Of candidat) = getListeCandidats()
            If tab.Count = 0 Then
                MsgBox("Vous ne pouvez pas accéder à ce menu s'il n'y a pas encore
                d'inscrits", vbOK)
            Else
                Me.Visible = False
                ModifSuppr.Visible = True
            End If
        End Sub
    '@brief : Ouvre le formulaire pour afficher la liste des inscrits (s'il y en a)
    Private Sub ButAffInscri_Click(sender As Object, e As EventArgs) Handles
        ButAffInscri.Click
            Dim tab As List(Of candidat) = getListeCandidats()
            If tab.Count = 0 Then
                MsgBox("Vous ne pouvez pas accéder à ce menu s'il n'y a pas encore
                d'inscrits", vbOK)
            Else
                Me.Visible = False
                AffInscrip.Visible = True
            End If
        End Sub
    '@brief : Ouvre le formulaire pour le bilan individuel ou par matière (s'il y a
    des candidats)
    Private Sub ButtonBilan_Click_1(sender As Object, e As EventArgs) Handles
        ButBilan.Click
            Dim tab As List(Of candidat) = getListeCandidats()
            If tab.Count = 0 Then
                MsgBox("Vous ne pouvez pas accéder à ce menu s'il n'y a pas encore
                d'inscrits", vbOK)
            Else
                Me.Visible = False
                Bilan.Visible = True
            End If
        End Sub
End Class
```

```
'@brief : Quitter l'application sans avoir fini les inscriptions
'      On crée une sauvegarde des candidats inscrits et du compteur d'inscrits
Private Sub ButQuitter_Click(sender As Object, e As EventArgs) Handles
ButQuitter.Click
    If Bilan.GetEstFinal Then
        Application.Exit()
    Else
        sauvegarder("sauvegarde.bin", "nbCpt.bin")
        Application.Exit()
    End If
End Sub
End Class
```

• InfosCandidat.vb

```
Public Class InfosCandidat
    'âge minimum proposé
    Private Const MIN_AGE As Integer = 18
    'âge maximum proposé
    Private Const MAX_AGE As Integer = 55
    'Taille maximale d'un code postal
    Private Const TAILLE_CP As Integer = 5
    'Délai de chaque tick de l'horloge
    Private Const DELAI As Integer = 1000
    'Initialise le compte à rebours
    Private chrono As Integer = 60

    '@brief : Au démarrage de ce formulaire, on met à jour les éléments,
    '           on paramètre et on lance le timer
    Private Sub InfosCandidat_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        SBAge.Minimum = MIN_AGE
        SBAge.Maximum = MAX_AGE
        SBAge.SmallChange = 1
        SBAge.LargeChange = 1
        txtAge.Enabled = False
        Timer1Min.Interval = DELAI
        Timer1Min.Start()
    End Sub
    '@brief : Met à jour le compte à rebours et l'heure actuelle
    '           Si le compte à rebours atteint 0, retourne à l'accueil
    Private Sub Timer() Handles Timer1Min.Tick
        Dim str As String = chrono
        Me.Text = TimeString + " Temps restant : " + str + " seconde(s)"
        If chrono = 0 Then
            Timer1Min.Stop()
            MsgBox("Vous avez dépassé la limite de temps.", "Temps écoulé")
            Me.Close()
            Accueil.Visible = True
        End If
        chrono -= 1
    End Sub
    '@brief : Empêche l'entrée de caractères non alphabétiques pour le nom, le prénom,
    et la ville
    Private Sub TxtAlpha_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
txtNom.KeyPress, txtPrenom.KeyPress, txtVille.KeyPress
        If e.KeyChar = vbBack Or e.KeyChar = " " Then
            Exit Sub
        End If
        e.Handled = Not Char.IsLetter(e.KeyChar) ' on écrit que des lettres de
l'alphabet
    End Sub
    '@brief : Empêche l'entrée de caractères alphabétiques pour le code postal
    Private Sub TxtCP_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
txtCP.KeyPress
        If e.KeyChar = vbBack Then
            Exit Sub
        End If
        If Not (e.KeyChar >= "0" AndAlso e.KeyChar <= "9") Then
            e.Handled = True 'On écrit que les caractères numériques
        End If
    End Sub
End Class
```

```

End Sub
'@brief : Empêche l'entrée de caractères non alphanumériques pour l'adresse
Private Sub TxtAdresse_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
txtAdresse.KeyPress
    If e.KeyChar = vbBack Or e.KeyChar = " " Then
        Exit Sub
    End If
    e.Handled = Not Char.IsLetterOrDigit(e.KeyChar) ' alphanumérique
End Sub
'@brief : Rend la saisie de l'utilisateur avec une bonne casse
Private Sub TxtAlpha_lostFocus(sender As Object, e As EventArgs) Handles
txtNom.LostFocus, txtPrenom.LostFocus, txtVille.LostFocus
    sender.Text = StrConv(sender.Text, vbProperCase)
End Sub
'@brief : Met à jour l'âge du candidat à chaque nouvelle valeur
Private Sub SBAge_Scroll(sender As Object, e As ScrollEventArgs) Handles
SBAge.Scroll
    txtAge.Text = SBAge.Value
End Sub
'@brief : Ferme le formulaire et retourne à l'accueil
Private Sub ButAban_Click(sender As Object, e As EventArgs) Handles ButAban.Click
    Me.Close()
    Accueil.Visible = True
End Sub
'@brief : Accède à la deuxième partie de l'inscription si tout est rempli
correctement
Private Sub ButSuite_Click(sender As Object, e As EventArgs) Handles
ButSuite.Click
    If String.IsNullOrEmpty(txtNom.Text) Then
        LabNom.ForeColor = Color.Red
    Else
        LabNom.ForeColor = Color.Black
    End If
    If String.IsNullOrEmpty(txtPrenom.Text) Then
        labPrenom.ForeColor = Color.Red
    Else
        labPrenom.ForeColor = Color.Black
    End If
    If String.IsNullOrEmpty(txtVille.Text) Then
        labVille.ForeColor = Color.Red
    Else
        labVille.ForeColor = Color.Black
    End If
    If String.IsNullOrEmpty(txtAge.Text) Then
        labAge.ForeColor = Color.Red
    Else
        labAge.ForeColor = Color.Black
    End If

    If String.IsNullOrEmpty(txtCP.Text) Or txtCP.TextLength <> TAILLE_CP Then
        labCP.ForeColor = Color.Red
    ElseIf txtCP.TextLength = TAILLE_CP Then
        If (txtCP.Text.Substring(0, 2) = "00") Then
            labCP.ForeColor = Color.Red
        Else
            labCP.ForeColor = Color.Black
        End If
    End If

```



```
End If

If LabNom.ForeColor = Color.Black And labPrenom.ForeColor = Color.Black And
labVille.ForeColor = Color.Black And
    labAge.ForeColor = Color.Black And labCP.ForeColor = Color.Black Then
    Me.Visible = False
    ChoixEpreuves.Visible = True
    Timer1Min.Stop()
End If

End Sub

End Class
```

• ChoixEpreuves.vb

```
Public Class ChoixEpreuves
    'Nombre d'épreuves écrites obligatoire
    Private Const NB_OBLI_ECRIT As Integer = 4
    'Nombre d'épreuves orales obligatoire
    Private Const NB_OBLI_ORAL As Integer = 3
    'Temps de départ du compte à rebours
    Private Const TMP_CHRONO As Integer = 90
    'Délai entre chaque tick de l'horloge
    Private Const DELAI As Integer = 1000
    'Nombre d'épreuves écrites coché
    Private nbCocheEcrit As Integer = 0
    'Nombre d'épreuves orales coché
    Private nbCocheOral As Integer = 0
    'Initialise le compte à rebours
    Private chrono As Integer = TMP_CHRONO
    'État du formulaire, inscription ou modification
    Private estInscription As Boolean
    'Tableau des noms de région proposés
    Private ReadOnly nomRegion() As String = {"Auvergne", "Bordelais", "Bourgogne",
    "Bretagne", "Corse", "Nord",
    "Normandie", "Paris", "Poitou", "Roussillon"}
    'Tableau des noms de matières écrites proposés
    Private ReadOnly nomEcrit() As String = {"Algorithmique", "Base de données",
    "Droit", "Expression", "Gestion", "Langage C",
    "Langage Java", "Mathématiques", "Programmation web", "Réseau", "Système",
    "Visual Basic .NET"}
    'Tableau des noms de matières orales proposés
    Private ReadOnly nomOral() As String = {"Droit", "Expression", "Gestion",
    "Mathématiques",
    "Réseau", "Système", "Anglais", "Chinois", "Espagnol"}

    '@brief : Setter de l'attribut estInscription
    Public Sub SetEstInscription(bool As Boolean)
        estInscription = bool
    End Sub
    '@brief : Initialise le formulaire (liste des régions, liste des matières et le
    timer)
    Private Sub ChoixEpreuves_Load(sender As Object, e As EventArgs) Handles
    MyBase.Load
        estInscription = True
        LabNom.Text = InfosCandidat.txtNom.Text
        labPrenom.Text = InfosCandidat.txtPrenom.Text
        RadButNon.Checked = True 'On coche le radio bouton Non par défaut

        For Each str As String In nomRegion
            cbxReg.Items.Add(str)
        Next
        cbxReg.SelectedIndex = 0
        ' Permet de faire des suggestions si l'utilisateur saisi
        cbxReg.DropDownStyle = ComboBoxStyle.DropDown
        cbxReg.AutoCompleteMode = AutoCompleteMode.Suggest
        cbxReg.AutoCompleteSource = AutoCompleteSource.ListItems

        Dim i As Integer
        For i = 0 To gboxEcrit.Controls.Count - 1
            If TypeOf gboxEcrit.Controls(i) Is CheckBox Then
```

```

        gboxEcrit.Controls(i).Text = nomEcrit(i)
    End If
Next
labNbEcrit.Text = NB_OBLI_ECRIT

For i = 0 To gboxOral.Controls.Count - 1
    If TypeOf gboxOral.Controls(i) Is CheckBox Then
        gboxOral.Controls(i).Text = nomOral(i)
    End If
Next
labNbOral.Text = NB_OBLI_ORAL

Timer1Min30.Interval = DELAI
Timer1Min30.Start()
End Sub
'@brief : Met à jour le compteur des épreuves écrites pour chaque checkbox cochée
ou décochée
Private Sub CheckEcrit_CheckedChanged(sender As Object, e As EventArgs) Handles
checkAlgo.CheckedChanged, checkBD.CheckedChanged,
    checkDroit.CheckedChanged, checkExp.CheckedChanged,
checkGest.CheckedChanged, checkC.CheckedChanged, checkJava.CheckedChanged,
    checkMaths.CheckedChanged, checkWeb.CheckedChanged,
checkRes.CheckedChanged, checkSys.CheckedChanged, checkVB.CheckedChanged

    If sender.Checked Then
        nbCocheEcrit += 1
        If nbCocheEcrit > NB_OBLI_ECRIT Then
            Dim nbEcrit As String = NB_OBLI_ECRIT
            MsgBox("Impossible de sélectionner plus de " + nbEcrit + " épreuves à
l'écrit", vbOK, "Limite de choix")
            sender.Checked = False
        End If
    Else nbCocheEcrit -= 1
End If

labNbEcrit.Text = NB_OBLI_ECRIT - nbCocheEcrit
CheckIndispo(sender, gboxOral)

End Sub
'@brief : Met à jour le compteur des épreuves orales pour chaque checkbox cochée
ou décochée
Private Sub CheckOral_CheckedChanged(sender As Object, e As EventArgs) Handles
checkOMaths.CheckedChanged,
    checkORes.CheckedChanged, checkOSys.CheckedChanged, checkOAng.CheckedChanged,
checkOChi.CheckedChanged,
    checkOEsp.CheckedChanged, checkODroit.CheckedChanged,
checkOExp.CheckedChanged, checkOGest.CheckedChanged

    If sender.Checked Then
        nbCocheOral += 1
        If nbCocheOral > NB_OBLI_ORAL Then
            Dim nbOral As String = NB_OBLI_ORAL
            MsgBox("Impossible de sélectionner plus de " + nbOral + " épreuves à
l'oral", vbOK, "Limite de choix")
            sender.Checked = False
        End If
    Else nbCocheOral -= 1
End If

```

```

labNbOral.Text = NB_OBLI_ORAL - nbCocheOral
CheckIndispo(sender, gboxEcrit)
End Sub
'@brief : Rend indisponible l'équivalent dans l'autre groupbox pour la matière
sélectionnée (pour une matière qui se situe à la fois dans Écrit et dans Oral)
Private Sub CheckIndispo(sender As Object, gbox As GroupBox)
    For Each str As String In nomOral
        For Each check As CheckBox In gbox.Controls
            If sender.Text = str And sender.Text = check.Text Then
                check.Enabled = Not sender.Checked
            End If
        Next
    Next
End Sub
'@brief : Fait apparaître le choix d'une épreuve facultative
Private Sub EpreuveFac() Handles labNbOral.TextChanged, labNbEcrit.TextChanged
    Dim condition As Boolean = labNbOral.Text = "0" AndAlso labNbEcrit.Text = "0"
    labQuest.Visible = condition
    RadButNon.Visible = condition
    RadButOui.Visible = condition
    cboxFac.Visible = False
End Sub
'@brief : Lorsque le bouton Oui (épreuve facultative) est cochée, on affiche la
liste des matières pour laquelle le candidat peut passer l'épreuve
Private Sub RadButOui_CheckedChanged(sender As Object, e As EventArgs) Handles
RadButOui.CheckedChanged
    cboxFac.Visible = RadButOui.Checked
    gboxEcrit.Enabled = Not RadButOui.Checked
    gboxOral.Enabled = Not RadButOui.Checked
    If RadButOui.Checked Then
        InitCboxFac()
    Else
        cboxFac.Items.Clear()
    End If
End Sub
'@brief : Initialise la combobox qui prend toutes les matières possibles
' pour lesquels le candidat peut choisir de passer pour l'épreuve
facultative
Private Sub InitCboxFac()
    For Each check As CheckBox In gboxOral.Controls
        If Not check.Checked Then
            'Si le nom appartient à un nom dans écrit également
            If nomEcrit.Contains(check.Text) Then
                For i As Integer = 0 To gboxEcrit.Controls.Count - 1
                    If TypeOf gboxEcrit.Controls(i) Is CheckBox Then
                        If gboxEcrit.Controls(i).Text = check.Text Then
                            Dim c As CheckBox = CType(gboxEcrit.Controls(i),
CheckBox)

                            If Not c.Checked Then
                                cboxFac.Items.Add(check.Text)
                            End If
                        End If
                    End If
                Next
            Else 'Si le nom est simplement dans le groupebox Oral
                cboxFac.Items.Add(check.Text)
            End If
        End If
    Next
End Sub

```

```

        End If
    Next
    cboxFac.SelectedIndex = 0
End Sub
'@brief : Ferme les formulaires liés à l'inscription
'      Si nous sommes dans une modification on retourne au formulaire ModifSuppr
'      Sinon retour à l'accueil
Private Sub FinChoixEpreuves()
    InfosCandidat.Close()
    Me.Close()
    If Not estInscription Then
        ModifSuppr.Show()
    Else
        Accueil.Visible = True
    End If
End Sub
'@brief : Lance la fin de l'inscription/modification
Private Sub ButAban_Click(sender As Object, e As EventArgs) Handles ButAban.Click
    FinChoixEpreuves()
End Sub
'@brief : Met à jour le compte à rebours et l'heure actuelle
'      Si le compte à rebours atteint 0, lance la fin de l'inscription
Private Sub Timer() Handles Timer1Min30.Tick
    Dim str As String = chrono
    Me.Text = TimeString + " Temps restant : " + str + " seconde(s)"
    If chrono = 0 Then
        Timer1Min30.Stop()
        MsgBox("Vous avez dépassé la limite de temps.",, "Limite de temps")
        FinChoixEpreuves()
    End If
    chrono -= 1
End Sub
'@brief : Permet d'accéder au récapitulatif s'il n'y a pas d'erreurs,
'      sinon, affiche un message lié à l'erreur
Private Sub ButEnreg_Click(sender As Object, e As EventArgs) Handles
ButEnreg.Click
    If nbCocheEcrit <> NB_OBLI_ECRIT Or nbCocheOral <> NB_OBLI_ORAL Then
        MsgBox("Vous n'avez pas coché le nombre d'épreuves réglementaire.", vbOK,
"Épreuves manquants")
    ElseIf cboxFac.SelectedIndex = -1 Then
        MsgBox("La région est incorrecte", vbOKOnly, "Région non définie")
    Else
        If Not estInscription Then 'Si nous sommes dans une modification, l'état
du formulaire Recap change
            Recap.SetEtat(Recap.EtatEnum.Modifier)
        End If
        Recap.Visible = True
        Me.Visible = False
        Timer1Min30.Stop() 'Le chrono ne tourne plus
        chrono = TMP_CHRONO ' On le réinitialise
    End If
End Sub
End Sub
End Class

```

• Recap.vb

```
Public Class Recap
    'Les différents états de ce formulaire
    Enum EtatEnum
        Enregistrer
        Modifier
        Supprimer
        Individuel
    End Enum
    'État actuel du formulaire
    Private etat As EtatEnum = EtatEnum.Enregistrer

    '@brief : Setter pour modifier l'attribut etat
    Public Sub SetEtat(e As EtatEnum)
        etat = e
    End Sub

    '@brief : Initialise le formulaire (état + informations du candidat + choix des matières)
    Private Sub Recap_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        If etat = EtatEnum.Enregistrer Then
            Titre.Text = "Récapitulatif"
            ButEnreg.Text = "Inscrire"
            ButRet.Visible = True
        ElseIf etat = EtatEnum.Modifier Then
            Titre.Text = "Modification"
            ButEnreg.Text = "Modifier"
            ButRet.Visible = True
        ElseIf etat = EtatEnum.Supprimer Then
            Titre.Text = "Suppression"
            ButRet.Visible = False
            ButEnreg.Text = "Supprimer"
        Else
            Titre.Text = "Bilan individuel"
            ButRet.Visible = False
            ButAban.Visible = False
            ButEnreg.BackColor = Color.White
            ButEnreg.Text = "Retour"
        End If

        'Partie informations sur le candidat
        txtNom.Text = InfosCandidat.txtNom.Text
        txtPrenom.Text = InfosCandidat.txtPrenom.Text
        txtCP.Text = InfosCandidat.txtCP.Text
        txtVille.Text = InfosCandidat.txtVille.Text
        txtAge.Text = InfosCandidat.txtAge.Text
        If Not Trim(InfosCandidat.txtAdresse.Text) = "" Then
            txtAdresse.Text = InfosCandidat.txtAdresse.Text
        Else
            txtAdresse.Enabled = False
        End If

        'Partie sur les choix des matières des épreuves du candidat
        txtReg.Text = ChoixEpreuves.cboReg.Text
        For Each check As CheckBox In ChoixEpreuves.gboxEcrit.Controls
            If check.Checked AndAlso lboxEpEcrit.Items.IndexOf(check) = -1 Then
                lboxEpEcrit.Items.Add(check.Text)
            End If
        End For
    End Sub
End Class
```

```

        End If
    Next
    For Each check As CheckBox In ChoixEpreuves.gboxOral.Controls
        If check.Checked AndAlso lboxEpOral.Items.IndexOf(check) = -1 Then
            lboxEpOral.Items.Add(check.Text)
        End If
    Next
    If ChoixEpreuves.RadButOui.Checked Then
        txtFac.Text = ChoixEpreuves.cboxFac.Text
    Else
        txtFac.Enabled = False
    End If
End Sub
'@brief : Ferme les formulaires liés à l'inscription
'         Si nous sommes dans un enregistrement on retourne à l'accueil
'         Si nous sommes dans une modification/suppression retour au formulaire
ModifSuppr
Private Sub FinRecap()
    Me.Close()
    InfosCandidat.Close()
    ChoixEpreuves.Close()
    If etat = EtatEnum.Enregistrer Then
        Accueil.Visible = True
    ElseIf etat = EtatEnum.Modifier Or etat = EtatEnum.Supprimer Then
        ModifSuppr.Show()
    End If
End Sub
'@brief : Lance la fin de l'inscription/modification/suppression
Private Sub ButAban_Click(sender As Object, e As EventArgs) Handles ButAban.Click
    FinRecap()
End Sub
'@brief : Retourne à la dernière partie de l'inscription
Private Sub ButRet_Click(sender As Object, e As EventArgs) Handles ButRet.Click
    Me.Close()
    ChoixEpreuves.Visible = True
    ChoixEpreuves.Timer1Min30.Start() 'On relance le timer
End Sub
'@brief : En fonction de l'attribut etat :
'         1) Enregistrer : ajoute un candidat à la liste des candidats
'         2) Modifier : modifie les informations du candidat
'         3) Supprimer : supprime le candidat de la liste
'         4) Individuel : retourne au formulaire Bilan
Private Sub ButEnreg_Click(sender As Object, e As EventArgs) Handles ButEnreg.Click
    Dim c As candidat
    'Initialisation du candidat
    c.nom = txtNom.Text
    c.prenom = txtPrenom.Text
    c.cp = txtCP.Text
    c.ville = txtVille.Text
    c.age = txtAge.Text
    c.region = txtReg.Text
    If txtAdresse.Enabled Then
        c.ad = txtAdresse.Text
    End If
    If txtFac.Enabled Then
        c.epFac = txtFac.Text
    End If
    ReDim c.tabEpreuvesEcrit(lboxEpEcrit.Items.Count - 1)

```

```

For i As Integer = 0 To lboxEpEcrit.Items.Count - 1
    c.tabEpreuvesEcrit(i) = lboxEpEcrit.Items(i)
Next
ReDim c.tabEpreuvesOral(lboxEpOral.Items.Count - 1)
For i As Integer = 0 To lboxEpOral.Items.Count - 1
    c.tabEpreuvesOral(i) = lboxEpOral.Items(i)
Next

If etat = EtatEnum.Enregistrer Then
    Ajouter(c)
ElseIf etat = EtatEnum.Modifier Then
    Modifier(c, ModifSuppr.getIndex())
ElseIf etat = EtatEnum.Supprimer Then
    Supprimer(ModifSuppr.getIndex)
Else
    Bilan.Visible = True
End If

FinRecap()
End Sub

'@brief : Empêche la saisie sur ce formulaire récapitulatif
Private Sub Txt_KeyPress(sender As Object, e As
System.Windows.Forms.KeyPressEventArgs) Handles txtNom.KeyPress, txtPrenom.KeyPress,
    txtAdresse.KeyPress, txtCP.KeyPress, txtVille.KeyPress, txtAge.KeyPress,
txtReg.KeyPress, txtFac.KeyPress
    e.Handled = True
End Sub
End Class

```


• ModifSuppr.vb

```
Public Class ModifSuppr
    'Indice à laquelle se trouve le candidat dans la liste
    Private Shared index As Integer
    '@brief : Getter, retourne l'index
    Public Function getIndex()
        Return index
    End Function
    '@brief : Initialise le formulaire (boutons + combobox des noms de candidat)
    Private Sub ModifSuppr_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        RadNum.Checked = True
        txtNum.Visible = True
        cboId.Visible = False

        Dim tab As List(Of candidat)
        tab = getListeCandidats()
        For i As Integer = 0 To tab.Count - 1
            cboId.Items.Add(tab(i).num & " " & Strings.UCase(tab(i).nom) & " " &
tab(i).prenom)
        Next
    End Sub
    '@brief : Contrôle la visibilité de txtNum
    Private Sub RadNum_CheckedChanged(sender As Object, e As EventArgs) Handles
RadNum.CheckedChanged
        txtNum.Visible = RadNum.Checked
    End Sub
    '@brief : Contrôle la visibilité de cboId
    Private Sub RadId_CheckedChanged(sender As Object, e As EventArgs) Handles
RadId.CheckedChanged
        cboId.Visible = RadId.Checked
    End Sub

    '@brief : Re-colore le fond en blanc pour signaler qu'un candidat a bien été
indiqué par l'utilisateur
    Private Sub RadCboxTextChanged(sender As Object, e As EventArgs) Handles
txtNum.TextChanged,
cboId.SelectedIndexChanged
        sender.BackColor = Color.White
    End Sub
    '@brief : Empêche l'entrée des caractères alphabétiques pour le numéro
    Private Sub TxtNum_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
txtNum.KeyPress
        If e.KeyChar = vbBack Then
            Exit Sub
        End If
        If Not (e.KeyChar >= "0" AndAlso e.KeyChar <= "9") Then
            e.Handled = True 'On écrit que les caractères numériques
        End If
    End Sub
    '@brief : Initialise le formulaire Recap pour une suppression d'un candidat
    Private Sub ButSuppr_Click(sender As Object, e As EventArgs) Handles
ButSuppr.Click
        InitForm(Recap.EtatEnum.Supprimer)
    End Sub
    '@brief : Initialise le formulaire Recap pour une modification des données d'un
candidat
```

```

Private Sub ButModif_Click(sender As Object, e As EventArgs) Handles
ButModif.Click
    InitForm(Recap.EtatEnum.Modifier)
End Sub
'@brief : Initialise le formulaire Recap
Private Sub InitForm(etat As Recap.EtatEnum)
    Dim num As Integer = 0
    Dim liste As List(Of candidat) = getListeCandidats()
    'Vérifie qu'un numéro ou un candidat a été sélectionné
    If RadNum.Checked And Not String.IsNullOrEmpty(txtNum.Text) Then
        For i As Integer = 0 To liste.Count - 1
            If txtNum.Text = liste(i).num Then
                num = txtNum.Text
            End If
        Next
        If num = 0 Then
            txtNum.BackColor = Color.LightGray
        End If

    ElseIf RadId.Checked And cboId.SelectedIndex <> -1 Then
        Dim indice As Integer
        indice = InStr(cboId.SelectedItem, " ")
        num = cboId.SelectedItem.Substring(0, indice)
    ElseIf RadNum.Checked Then
        txtNum.BackColor = Color.LightGray
    Else
        cboId.BackColor = Color.LightGray
    End If

    If num <> 0 Then
        For i As Integer = 0 To liste.Count - 1
            If liste(i).num = num Then
                index = i
            End If
        Next

        'Initialise les données du formulaire InfosCandidat
        InfosCandidat.txtNom.Text = liste(index).nom
        InfosCandidat.txtPrenom.Text = liste(index).prenom
        InfosCandidat.txtAdresse.Text = liste(index).ad
        InfosCandidat.txtVille.Text = liste(index).ville
        InfosCandidat.txtCP.Text = liste(index).cp
        InfosCandidat.txtAge.Text = liste(index).age

        Recap.SetEtat(etat)
        If etat = Recap.EtatEnum.Modifier Then
            ChoixEpreuves.Show()
            ChoixEpreuves.SetEstInscription(False)
            Dim idxReg As Integer
            idxReg = ChoixEpreuves.cboReg.Items.IndexOf(liste(index).region)
            ChoixEpreuves.cboReg.SelectedIndex = idxReg

            For Each tabEcrit As String In liste(index).tabEpreuvesEcrit
                For Each check As CheckBox In ChoixEpreuves.gboxEcrit.Controls
                    If check.Text = tabEcrit Then
                        check.Checked = True
                    End If
                Next
            Next
        End If
    End If
End Sub

```

```

        Next
    Next
    For Each tabOral As String In liste(index).tabEpreuvesOral
        For Each check As CheckBox In ChoixEpreuves.gboxOral.Controls
            If check.Text = tabOral Then
                check.Checked = True
            End If
        Next
    Next
    If liste(index).epFac <> "" Then
        ChoixEpreuves.RadButOui.Checked = True
        Dim idxFac As Integer
        idxFac = ChoixEpreuves.cboxFac.Items.IndexOf(liste(index).epFac)
        ChoixEpreuves.cboxFac.SelectedIndex = idxFac
    End If

    ElseIf etat = Recap.EtatEnum.Supprimer Then
        Recap.Show()
        Recap.txtReg.Text = liste(index).region
        Recap.txtFac.Text = liste(index).epFac

        For i As Integer = 0 To liste(index).tabEpreuvesEcrit.Length - 1
            Recap.lboxEpEcrit.Items.Add(liste(index).tabEpreuvesEcrit(i))
        Next
        For i As Integer = 0 To liste(index).tabEpreuvesOral.Length - 1
            Recap.lboxEpOral.Items.Add(liste(index).tabEpreuvesOral(i))
        Next

    End If
    Me.Close()
End If
End Sub
'@brief : Retourne à l'accueil
Private Sub ButRet_Click(sender As Object, e As EventArgs) Handles ButRet.Click
    Me.Close()
    Accueil.Show()
End Sub
End Class

```

- AffInscrip.vb

Public Class AffInscrip

```
'@brief : Initialise le formulaire et ses ListBox par ordre alphabétique des noms
Private Sub AffInscrip_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Dim tab As List(Of candidat) = getListeCandidats()
    tab = tab.OrderBy(Function(c) c.nom).ThenBy(Function(c) c.prenom).ToList
    For Each c As candidat In tab
        lboxNum.Items.Add(c.num)
        lboxNom.Items.Add(c.nom)
        lboxPrenom.Items.Add(c.prenom)
        Dim nomEcrit As String = ""
        For i As Integer = 0 To c.tabEpreuvesEcrit.Length - 1
            If c.tabEpreuvesEcrit(i) = "Langage C" Or c.tabEpreuvesEcrit(i) =
"Langage Java" Then
                Dim indexEspace = c.tabEpreuvesEcrit(i).IndexOf(" ")
                nomEcrit += c.tabEpreuvesEcrit(i).Substring(0, 2) &
c.tabEpreuvesEcrit(i).Substring(indexEspace + 1, 1) & " "
            Else
                nomEcrit += c.tabEpreuvesEcrit(i).Substring(0, 3) & " "
            End If
        Next
        lboxEcrit.Items.Add(Strings.UCase(nomEcrit))
        Dim nomOral As String = ""
        For i As Integer = 0 To c.tabEpreuvesOral.Length - 1
            nomOral += c.tabEpreuvesOral(i).Substring(0, 3) & " "
        Next
        lboxOral.Items.Add(Strings.UCase(nomOral))
        Dim str As String = c.epFac
        If String.IsNullOrEmpty(c.epFac) Then
            str = ""
        End If
        lboxFac.Items.Add(str)
    Next

    SBInscripts.Maximum = tab.Count - 1

End Sub
'@brief : Synchronise la sélection d'un candidat pour les 6 ListBox
Private Sub ListBox_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
lboxNum.SelectedIndexChanged, lboxNom.SelectedIndexChanged,
    lboxPrenom.SelectedIndexChanged, lboxEcrit.SelectedIndexChanged,
lboxOral.SelectedIndexChanged, lboxFac.SelectedIndexChanged

    Dim index As Integer = sender.selectedIndex
    For Each ctrl As Control In pnlList.Controls
        If TypeOf ctrl Is ListBox Then
            DirectCast(ctrl, ListBox).SelectedIndex = index
            sender.topIndex = DirectCast(ctrl, ListBox).TopIndex
            SBInscripts.Value = DirectCast(ctrl, ListBox).SelectedIndex
        End If
    Next
End Sub
'@brief : Retourne à l'accueil
Private Sub ButRet_Click(sender As Object, e As EventArgs) Handles ButRet.Click
    Me.Close()
    Accueil.Visible = True
```

```

End Sub

'@brief : Change l'index de toutes les ListBox lors du défilement de la barre
verticale
Private Sub SBInscrits_Scroll(sender As Object, e As ScrollEventArgs) Handles
SBInscrits.Scroll
    For Each ctrl As Control In pnlList.Controls
        If TypeOf ctrl Is ListBox Then
            DirectCast(ctrl, ListBox).SelectedIndex = SBInscrits.Value
        End If
    Next
End Sub
'@brief : Empêche le défilement vertical des ListBox
Private Sub ListBox_MouseWheel(sender As Object, e As MouseEventArgs) Handles
lboxNum.MouseWheel, lboxNom.MouseWheel,
    lboxPrenom.MouseWheel, lboxEcrit.MouseWheel, lboxOral.MouseWheel,
lboxFac.MouseWheel
    Dim mouseEvent As HandledMouseEventArgs = DirectCast(e, HandledMouseEventArgs)
    mouseEvent.Handled = True
End Sub
End Class

```

• Bilan.vb

Imports System.IO

Public Class Bilan

'Indique si les inscriptions sont terminées

Private estFinal As Boolean

'Tableau des noms de matières écrites

Private ReadOnly nomsMatiere() As String = {"Algorithmique", "Base de données", "Droit", "Expression", "Gestion", "Langage C", "Langage Java", "Mathématiques", "Programmation web", "Réseau", "Système", "Visual Basic .NET", "Anglais", "Chinois", "Espagnol"}

'Tableau des noms de matières orales

Private ReadOnly nomRegion() As String = {"Auvergne", "Bordelais", "Bourgogne", "Bretagne", "Corse", "Nord", "Normandie", "Paris", "Poitou", "Roussillon"}

'@brief : Getter pour l'attribut estFinal

Public Function GetEstFinal()

Return estFinal

End Function

'@brief : Initialise le formulaire (estFinal + les deux combobox)

Private Sub Bilan_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Dim vbVal As Integer

vbVal = MsgBox("Avez-vous terminé d'enregistrer les inscriptions ?", vbYesNo, "Bilan provisoire ou final")

If vbVal = vbYes Then

estFinal = True

Else

estFinal = False

End If

If estFinal Then

ButRet.Text = "Quitter"

Else

ButRet.Text = "Retour"

End If

RadIndiv.Checked = True

cboxIndiv.Visible = True

cboxMatiere.Visible = False

ButMatiere.Visible = False

Dim tab As List(Of candidat) = getListeCandidats()

txtNbInscrits.Text = tab.Count

For i As Integer = 0 To tab.Count - 1

cboxIndiv.Items.Add(Strings.UCase(tab(i).nom) & " " & tab(i).prenom)

Next

cboxIndiv.DropDownStyle = ComboBoxStyle.DropDown

cboxIndiv.AutoCompleteMode = AutoCompleteMode.Suggest

cboxIndiv.AutoCompleteSource = AutoCompleteSource.ListItems

For i = 0 To nomsMatiere.Length - 1

cboxMatiere.Items.Add(nomsMatiere(i))

Next

End Sub

```

'@brief : Empêche la saisie dans le nombre d'inscrits
Private Sub TxtNbInscrits_KeyPress(sender As Object, e As
System.Windows.Forms.KeyPressEventArgs) Handles txtNbInscrits.KeyPress
    e.Handled = True
End Sub
'@brief : Contrôle la visibilité de cboIndiv et ButIndiv
Private Sub RadIndiv_CheckedChanged(sender As Object, e As EventArgs) Handles
RadIndiv.CheckedChanged
    cboIndiv.Visible = RadIndiv.Checked
    ButIndiv.Visible = RadIndiv.Checked
End Sub
'@brief : Contrôle la visibilité de cboMatiere et ButMatiere
Private Sub RadMatiere_CheckedChanged(sender As Object, e As EventArgs) Handles
RadMatiere.CheckedChanged
    cboMatiere.Visible = RadMatiere.Checked
    ButMatiere.Visible = RadMatiere.Checked
End Sub
'@brief : Si les inscriptions sont terminées, crée les fichiers bilans en fonction
de la matière et de la région
'
    Sinon, retour à l'accueil
Private Sub ButRet_Click(sender As Object, e As EventArgs) Handles ButRet.Click
    If estFinal Then
        Dim tab As List(Of candidat) = getListeCandidats()
        tab = tab.OrderBy(Function(c) c.region).ThenBy(Function(c)
c.nom).ThenBy(Function(c) c.prenom).ToList
        Dim fichier As StreamWriter
        Dim aDesCandidatsOral As Boolean = False

        'Pour chaque matière et pour chaque région, crée un fichier texte s'il y a
des candidats inscrits
        For Each str As String In nomsMatiere
            For Each r As String In nomRegion
                For Each c As candidat In tab
                    If c.region = r Then
                        'Ajoute les candidats pour l'épreuve à l'écrit
                        For i As Integer = 0 To c.tabEpreuvesEcrit.Length - 1
                            If c.tabEpreuvesEcrit(i) = str Then
                                If Not File.Exists(Strings.UCase(r) & "-" &
Strings.UCase(str) & ".txt") Then
                                    fichier = New StreamWriter(Strings.UCase(r) &
 "-" & Strings.UCase(str) & ".txt")
                                    fichier.WriteLine("Liste des candidats de la
région " & r & " pour la matière " & str & " à l'écrit")
                                End If
                                fichier.WriteLine(c.nom & " " & c.prenom & " " &
c.num)
                            End If
                        Next
                    End If
                Next
            For Each c As Candidat In tab
                If c.region = r Then
                    'Ajoute les candidats pour l'épreuve à l'oral
                    For i As Integer = 0 To c.tabEpreuvesOral.Length - 1
                        If c.tabEpreuvesOral(i) = str Then
                            If Not File.Exists(Strings.UCase(r) & "-" &
Strings.UCase(str) & ".txt") Then

```

```

        fichier = New StreamWriter(Strings.UCase(r) &
        "-" & Strings.UCase(str) & ".txt")
        fichier.WriteLine("Liste des candidats de la
région " & r & " pour la matière " & str & " à l'oral")
        aDesCandidatsOral = True
    ElseIf Not aDesCandidatsOral Then
        fichier.WriteLine("Liste des candidats de la
région " & r & " pour la matière " & str & " à l'oral")
        aDesCandidatsOral = True
    End If
    fichier.WriteLine(c.nom & " " & c.prenom & " " &
c.num)
    End If
Next
'Ajoute les candidats pour l'épreuve en facultatif
If c.epFac = str Then
    If Not File.Exists(Strings.UCase(r) & "-" &
Strings.UCase(str) & ".txt") Then
        fichier = New StreamWriter(Strings.UCase(r) & "-"
& Strings.UCase(str) & ".txt")
        fichier.WriteLine("Liste des candidats de la
région " & r & " pour la matière " & str & " à l'oral")
        aDesCandidatsOral = True
    ElseIf Not aDesCandidatsOral Then
        fichier.WriteLine("Liste des candidats de la
région " & r & " pour la matière " & str & " à l'oral")
        aDesCandidatsOral = True
    End If
    fichier.WriteLine(c.nom & " " & c.prenom & " " &
c.num)
    End If
End If
Next
fichier.Close()
aDesCandidatsOral = False
Next
Next
'Supprime les sauvegardes des candidats s'il y en a
If File.Exists("sauvegarde.bin") And File.Exists("nbCpt.bin") Then
    File.Delete("sauvegarde.bin")
    File.Delete("nbCpt.bin")
End If
Application.Exit()
Else
    Me.Close()
    Accueil.Show()
End If
End Sub
'@brief : Initialise le formulaire Recap pour le bilan d'un candidat
Private Sub ButIndiv_Click(sender As Object, e As EventArgs) Handles
ButIndiv.Click
    Dim liste As List(Of candidat) = getlisteCandidats()
    Dim index As Integer
    Dim nom As String
    Dim prenom As String
    Dim item As String

    If cboIndiv.SelectedIndex <> -1 Then

```



```

item = cboIndiv.SelectedItem.ToString()
nom = item.Substring(0, item.IndexOf(" "))
prenom = item.Substring(item.IndexOf(" ") + 1)

For i As Integer = 0 To liste.Count - 1
    If liste(i).nom = nom And liste(i).prenom = prenom Then
        index = i
    End If
Next

InfosCandidat.txtNom.Text = liste(index).nom
InfosCandidat.txtPrenom.Text = liste(index).prenom
InfosCandidat.txtAdresse.Text = liste(index).ad
InfosCandidat.txtVille.Text = liste(index).ville
InfosCandidat.txtCP.Text = liste(index).cp
InfosCandidat.txtAge.Text = liste(index).age
Recap.SetEtat(Recap.EtatEnum.Individuel)

Recap.Show()
Recap.txtReg.Text = liste(index).region
Recap.txtFac.Text = liste(index).epFac
For i As Integer = 0 To liste(index).tabEpreuvesEcrit.Length - 1
    Recap.lboxEpEcrit.Items.Add(liste(index).tabEpreuvesEcrit(i))
Next
For i As Integer = 0 To liste(index).tabEpreuvesOral.Length - 1
    Recap.lboxEpOral.Items.Add(liste(index).tabEpreuvesOral(i))
Next
Me.Visible = False
Else
    cboIndiv.BackColor = Color.LightGray
End If
End Sub
'@brief : Accède au bilan des matières si une matière est sélectionnée
Private Sub ButMatiere_Click(sender As Object, e As EventArgs) Handles
ButMatiere.Click
    If cboMatiere.SelectedItem <> "" Then
        BilanMatiere.Show()
        Me.Visible = False
    Else
        cboMatiere.BackColor = Color.LightGray
    End If
End Sub

'@brief : Re-colore le fond en blanc pour signaler qu'un candidat a bien été
choisi par l'utilisateur
Private Sub cbo_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
cboIndiv.SelectedIndexChanged,
    cboMatiere.SelectedIndexChanged
    sender.BackColor = Color.White
End Sub
End Class

```

• BilanMatiere.vb

```
Public Class BilanMatiere
    '@brief : Initialise le formulaire (nom de la matière + les listBox + les barres
de défilement)
    Private Sub BilanMatiere_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        Dim tab As List(Of Candidat) = GetListeCandidats()
        txtMatiere.Text = Bilan.cboxMatiere.SelectedItem
        tab = tab.OrderBy(Function(c) c.region).ThenBy(Function(c)
c.nom).ThenBy(Function(c) c.prenom).ToList
        SB_Ecrit.Maximum = 0
        SB_Oral.Maximum = 0
        For Each c As candidat In tab
            For i As Integer = 0 To c.tabEpreuvesEcrit.Length - 1
                If c.tabEpreuvesEcrit(i) = txtMatiere.Text Then
                    lboxRegEcrit.Items.Add(c.region)
                    lboxNomEcrit.Items.Add(c.nom)
                    lboxPrenomEcrit.Items.Add(c.prenom)
                    SB_Ecrit.Maximum += 1
                End If
            Next
            For i As Integer = 0 To c.tabEpreuvesOral.Length - 1
                If c.tabEpreuvesOral(i) = txtMatiere.Text Then
                    lboxRegOral.Items.Add(c.region)
                    lboxNomOral.Items.Add(c.nom)
                    lboxPrenomOral.Items.Add(c.prenom)
                    SB_Oral.Maximum += 1
                End If
            Next
            If c.epFac = txtMatiere.Text Then
                lboxRegOral.Items.Add(c.region)
                lboxNomOral.Items.Add(c.nom)
                lboxPrenomOral.Items.Add(c.prenom)
                SB_Oral.Maximum += 1
            End If
        Next

        SB_Ecrit.Maximum -= 1
        SB_Oral.Maximum -= 1

    End Sub
    '@brief : Retourne à l'accueil
    Private Sub ButRet_Click(sender As Object, e As EventArgs) Handles ButRet.Click
        Me.Close()
        Bilan.Visible = True
    End Sub

    '@brief : Synchronise la sélection d'un candidat pour les 3 ListBox Écrit
    Private Sub LboxEcrit_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles lboxRegEcrit.SelectedIndexChanged,
        lboxPrenomEcrit.SelectedIndexChanged, lboxNomEcrit.SelectedIndexChanged

        Dim index As Integer = sender.selectedIndex
        For Each ctrl As Control In pnlEcrit.Controls
            If TypeOf ctrl Is ListBox Then
                DirectCast(ctrl, ListBox).SelectedIndex = index
                sender.topIndex = DirectCast(ctrl, ListBox).TopIndex
            End If
        Next
    End Sub
End Class
```

```

        SB_Ecrit.Value = DirectCast(ctrl, ListBox).SelectedIndex
    End If
Next
End Sub

'@brief : Synchronise la sélection d'un candidat pour les 3 ListBox Oral
Private Sub ListBoxOral_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
lboxRegOral.SelectedIndexChanged,
lboxPrenomOral.SelectedIndexChanged, lboxNomOral.SelectedIndexChanged

    Dim index As Integer = sender.selectedIndex
    For Each ctrl As Control In pnlOral.Controls
        If TypeOf ctrl Is ListBox Then
            DirectCast(ctrl, ListBox).SelectedIndex = index
            sender.topIndex = DirectCast(ctrl, ListBox).TopIndex
            SB_Oral.Value = DirectCast(ctrl, ListBox).SelectedIndex
        End If
    Next
End Sub

'@brief : Change l'index des 3 ListBox Ecrit lors du défilement de leur barre
verticale
Private Sub SB_Ecrit_Scroll(sender As Object, e As ScrollEventArgs) Handles
SB_Ecrit.Scroll
    For Each ctrl As Control In pnlEcrit.Controls
        If TypeOf ctrl Is ListBox Then
            DirectCast(ctrl, ListBox).SelectedIndex = SB_Ecrit.Value
        End If
    Next
End Sub

'@brief : Change l'index des 3 ListBox Oral lors du défilement de leur barre
verticale
Private Sub SBInscriits_Scroll(sender As Object, e As ScrollEventArgs) Handles
SB_Oral.Scroll
    For Each ctrl As Control In pnlOral.Controls
        If TypeOf ctrl Is ListBox Then
            DirectCast(ctrl, ListBox).SelectedIndex = SB_Oral.Value
        End If
    Next
End Sub

'@brief : Empêche le défilement vertical des ListBox
Private Sub ListBox_MouseWheel(sender As Object, e As MouseEventArgs) Handles
lboxRegEcrit.MouseWheel,
lboxPrenomEcrit.MouseWheel, lboxNomEcrit.MouseWheel, lboxRegOral.MouseWheel,
lboxPrenomOral.MouseWheel,
lboxNomOral.MouseWheel
    Dim mouseEvent As HandledMouseEventArgs = DirectCast(e, HandledMouseEventArgs)
    mouseEvent.Handled = True
End Sub

'@brief : Empêche d'écrire dans le nom de la matière
Private Sub txtMatiere_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
txtMatiere.KeyPress
    e.Handled = True
End Sub
End Class

```

Conclusion

Durant ce projet, nous avons essayé du mieux que nous pouvons de respecter tout le cahier des charges fourni. Nous sommes fiers du résultat de notre application, on a essayé de résoudre tous les bugs que nous avons trouvé et ne pas les laisser exister.

Toutefois, nous trouvons également quelques points d'améliorations possibles :

- Lors de la saisie des renseignements du candidat (InfosCandidat), il est indiqué dans l'énoncé que les champs Adresse et Ville soient alphanumérique et alphabétique. C'est ce que nous avons fait mais nous sommes un peu gênés de ne pas pouvoir mettre de trait d'union ou d'apostrophe, on pourrait mettre une condition seulement pour ces deux caractères mais ce n'est pas demandé.
- Pour l'affichage de la liste des inscrits (AffInscrip) et le bilan des matières (BilanMatière), on a décidé de cacher les barres de défilement de toutes les ListBox en les faisant « dépasser » les unes sur les autres, et de n'en utiliser qu'une pour en contrôler plusieurs ; C'est pour pouvoir les synchroniser en scrollant. Cependant, les barres de défilement existent toujours même si elles sont cachées, il aurait été mieux soit de les faire disparaître, soit de synchroniser les barres de chaque ListBox entre elles. Malheureusement pour la première solution nous avons plus ou moins réussi mais les barres de défilement apparaissaient toujours à certaines occasions, et nous n'avons pas trouvé comment faire pour la deuxième.

Pour y voir plus clair :

NUMÉRO	NOM	PRÉNOM	ÉCRIT	ORAL	FACULTATIVE
5	Ère	Axel	GES LAC MAT PRO	DRO RÉSP ESP	Anglais
8	Golay	Harry	EXP GES LAC PRO	DRO MAT RÉSP	Système
7	Golay	Jerry	BAS DRO MAT PRO	EXP RÉSP SYS	Chinois
4	Hyère	Claire	ALG BAS LAJ MAT	DRO EXP GES	Anglais
6	Kiroul	Pierre	GES LAC SYS VIS	ANG CHI ESP	Réseau
9	Masse	Sarah	ALG DRO MAT RÉSP	EXP SYS ESP	Gestion
3	Neymar	Jean	DRO EXP PRO RÉSP	ANG CHI ESP	Mathématiques

Figure 26. *Avant* - Liste des inscrits (Scrollbar cachés)

NUMÉRO	NOM	PRÉNOM	ÉCRIT	ORAL	FACULTATIVE
5	Ère	Axel	GES LAC MAT PRO	DRO RÉSP ESP	Anglais
8	Golay	Harry	EXP GES LAC PRO	DRO MAT RÉSP	Système
7	Golay	Jerry	BAS DRO MAT PRO	EXP RÉSP SYS	Chinois
4	Hyère	Claire	ALG BAS LAJ MAT	DRO EXP GES	Anglais
6	Kiroul	Pierre	GES LAC SYS VIS	ANG CHI ESP	Réseau
9	Masse	Sarah	ALG DRO MAT RÉSP	EXP SYS ESP	Gestion
3	Neymar	Jean	DRO EXP PRO RÉSP	ANG CHI ESP	Mathématiques

Figure 27. *Après* - Liste des inscrits (Scrollbar cachés par les listes)