

PREUVES

1.Interface Homme-machine.....	2
1.1.Je sais décrire le contexte de mon application pour qu'il soit compréhensible par tout le monde.....	2
1.2.Je sais dessiner des sketches pour concevoir les fenêtres de mon application.	2
1.3.Je sais enchaîner mes sketches au sein d'un story-board.....	2
1.4.Je sais concevoir un diagramme de cas d'utilisation qui représente les fonctionnalités de mon application.....	2
1.5.Je sais concevoir une application ergonomique.	2
1.6.Je sais concevoir une application avec une prise en compte de l'accessibilité.....	2
1.7.Je sais choisir mes layouts à bon escient.....	2
1.8.Je sais choisir mes composants à bon escient.	3
1.9.Je sais créer mon propre composant.	3
1.10.Je sais personnaliser mon application en utilisant des ressources et des styles.	4
1.11.Je sais utiliser les DataTemplate (locaux et globaux).	5
1.12.Je sais intercepter les évènements de la vue.	5
1.13.Je sais notifier la vue depuis des évènements métier.	5
1.14.Je sais gérer le DataBinding sur mon master.	5
1.15.Je sais gérer le DataBinding sur mon détail.	5
1.16.Je sais gérer le DataBinding et les Dependency Property sur mes UserControl.....	5
1.17.Je sais développer un Master/Detail.....	5
2.Conception et Programmation Orientées Objets.....	6
2.1.Je sais concevoir un diagramme de classes qui représente mon application.	6
2.2.Je sais réaliser un diagramme de paquetages qui illustre bien l'isolation entre les parties de mon application.....	6
2.3.Je sais décrire mes deux diagrammes en mettant en valeur et en justifiant les éléments essentiels.	6
2.4.Je maîtrise les bases de la programmation C# (classes, structures, instances...).	6
2.5.Je sais utiliser l'abstraction à bon escient (héritage, interfaces, polymorphisme).	6
2.6.Je sais gérer des collections simples (tableaux, listes.....)	6
2.7.Je sais gérer des collections avancées (dictionnaires).....	6
2.8.Je sais contrôler l'encapsulation au sein de mon application.....	7
2.9.Je sais tester mon application.....	7
2.10.Je sais utiliser LINQ.....	7
2.11.Je sais gérer les évènements.	7

3.Projet Tuteuré S2.....	7
3.1.Je sais mettre en avant dans mon diagramme de classes la persistance de mon application.	7
3.2.Je sais mettre en avant dans mon diagramme de classes ma partie personnelle.	8
3.3.Je sais mettre en avant dans mon diagramme de paquetages la persistance de mon application.	8
3.4.Je sais réaliser une vidéo de 1 à 3 minutes qui montre la démo de mon application.....	8
3.5.Je sais coder la persistance au sein de mon application.	8
3.6.Je sais coder une fonctionnalité qui m'est personnelle.	8
3.7.Je sais documenter mon code.	8
3.8.Je sais utiliser SVN.	8
3.9.Je sais développer une application qui compile.	8
3.10.Je sais développer une application fonctionnelle.	9
3.11.Je sais mettre à disposition un outil pour déployer mon application.	9

1. Interface Homme-machine

1.1. Je sais décrire le contexte de mon application pour qu'il soit compréhensible par tout le monde.

Cf. /documentation/IHM.pdf (pages 1 à 3).

1.2. Je sais dessiner des sketches pour concevoir les fenêtres de mon application.

Cf. /documentation/IHM.pdf (pages 4 à 6).

1.3. Je sais enchaîner mes sketches au sein d'un story-board.

Cf. /documentation/IHM.pdf (pages 7 à 12).

1.4. Je sais concevoir un diagramme de cas d'utilisation qui représente les fonctionnalités de mon application.

Cf. /documentation/IHM.pdf (page 13).

1.5. Je sais concevoir une application ergonomique.

1.6. Je sais concevoir une application avec une prise en compte de l'accessibilité.

1.7. Je sais choisir mes layouts à bon escient.

Master

```

70 <ListBox VerticalAlignment="Stretch" Background="black" x:Name="MasterDesLangages" MaxWidth="150" MinWidth="20" Padding="3,1"
71     SelectedItem="{Binding ItemsLangages.ItemCourant}" ItemsSource="{Binding ItemsLangages.LesItems}"
72     SelectionChanged="MasterDesLangages_SelectionChanged">
73     <ListBox.ItemTemplate>
74     <!--
75         Ce DataTemplate permet de contrôler l'affichage du
76         langage dans la ListBox et n'affiche que le nom du langage
77     -->
78     <DataTemplate>
79     <Border Background="white" MinWidth="90" MinHeight="30" CornerRadius="10">
80     <TextBlock Text="{Binding Nom}" FontWeight="Medium" FontSize="15" Foreground="Orange" Padding="2"
81         VerticalAlignment="Center" HorizontalAlignment="Center"/>
82     </Border>
83     </DataTemplate>
84     </ListBox.ItemTemplate>
85 </ListBox>

```

/source/repos/notre_solution/notre_projet/MainWindow.xaml ligne 70

Detail

```

<!--
    Cette grille contient les 3 sections d'un langage.
    C'est la grille qui à été retenu car elle elle s'adapte
    à son conteneur.
-->
<DockPanel DataContext="{Binding ItemsLangages.ItemCourant}">

```

/source/repos/notre_solution/notre_projet/DescriptionLangageUC.xaml ligne 14

1.8. Je sais choisir mes composants à bon escient.

```

<!--
    Cette ListBox contient tout les langages recensés par le logiciel.

    C'est la ListBox qui a été retenue car elle permet une gestion
    simple de l'affichage, récupération simple des données sauvgardées,
    ainsi qu'une récupération simple des données sélectionnées par
    l'utilisateur (le langage).

    Cette ListBox sera placé à gauche de l'application grâce au DockPanel
    qui occupe l'ensemble de le fenêtre. Elle occupera aussi toute la
    hauteur de la fenêtre (mode Stretch)
-->

```

```

<!--
    Cette grille contient les 3 sections d'un langage.
    C'est la grille qui à été retenu car elle elle s'adapte
    à son conteneur.
-->
<DockPanel DataContext="{Binding ItemsLangages.ItemCourant}">

```

/source/repos/notre_solution/notre_projet/MainWindow ligne 58

1.9. Je sais créer mon propre composant.

Cf. /source/repos/notre_solution/notre_projet/DescriptionLangageUC.xaml

1.10. Je sais personnaliser mon application en utilisant des ressources et des styles.

```
<Application.Resources>

  <!--
    Ce style s'applique au bordures des sections (Histoire, spécificité, utilité).
    C'est une bordure noire large de 2, avec un padding de 5 et une marge de 8
  -->
  <Style TargetType="Border" x:Key="sectionBorder">
    <Setter Property="BorderBrush" Value="Black"/>
    <Setter Property="BorderThickness" Value="2"/>
    <Setter Property="Padding" Value="5"/>
    <Setter Property="Margin" Value="8"/>
  </Style>

  <!--
    Ce style s'applique au bordure des elements des sections
    C'est une bordure noire, large de 1, avec un padding de 8
    et une marge de 4
  -->
  <Style TargetType="Border" x:Key="elementBorder">
    <Setter Property="BorderBrush" Value="Black"/>
    <Setter Property="BorderThickness" Value="1"/>
    <Setter Property="Padding" Value="8"/>
    <Setter Property="Margin" Value="4"/>
  </Style>

  <!--
    Ce style s'applique au titre des éléments des sections.
    Ces titres sont en gras et on une taille de police de 18
  -->
  <Style TargetType="TextBlock" x:Key="titreElementText">
    <Setter Property="FontWeight" Value="Bold"/>
    <Setter Property="FontSize" Value="18"/>
  </Style>

  <!--
    Ce style s'applique aux élément de la ListBox
  -->
  <Style TargetType="Button" x:Key="listBoxBouton">
    <Setter Property="Background" Value="White"/>
    <Setter Property="MinHeight" Value="35"/>
    <Setter Property="Padding" Value="5"/>
    <Setter Property="Width" Value="65"/>
  </Style>
</Application.Resources>
```

/source/repos/notre_solution/notre_projet/App.xaml (lignes 6 à 67)

1.11. Je sais utiliser les DataTemplate (locaux et globaux).

```

70 <ListBox VerticalAlignment="Stretch" Background="black" x:Name="MasterDesLangages" MaxWidth="150" MinWidth="20" Padding="3,1"
71     SelectedItem="{Binding ItemsLangages.ItemCourant}" ItemsSource="{Binding ItemsLangages.LesItems}"
72     SelectionChanged="MasterDesLangages_SelectionChanged">
73     <ListBox.ItemTemplate>
74     <!--
75         Ce DataTemplate permet de contrôler l'affichage du
76         langage dans la ListBox et n'affiche que le nom du langage
77     -->
78     <DataTemplate>

```

1.12. Je sais intercepter les événements de la vue.

```

<MenuItem Header="Connexion">
    <MenuItem Header="Se Connecter" Click="SeConnecter_Click"/>
    <Separator/>
    <MenuItem Header="Se Déconnecter" Click="Deconnexion_Click"/>
</MenuItem>

```

/source/repos/notre_solution/notre_projet/MainWindow.xml lignes 29, 31

1.13. Je sais notifier la vue depuis des événements métier.

```

70 <ListBox VerticalAlignment="Stretch" Background="black" x:Name="MasterDesLangages" MaxWidth="150" MinWidth="20" Padding="3,1"
71     SelectedItem="{Binding ItemsLangages.ItemCourant}" ItemsSource="{Binding ItemsLangages.LesItems}"
72     SelectionChanged="MasterDesLangages_SelectionChanged">
73     <ListBox.ItemTemplate>
74     <!--
75         Ce DataTemplate permet de contrôler l'affichage du
76         langage dans la ListBox et n'affiche que le nom du langage
77     -->
78     <DataTemplate>

```

Cf. /source/repos/notre_solution/notre_bibliotheque/Items.cs
(INotifyPropertyChanged)

1.14. Je sais gérer le DataBinding sur mon master.

```

70 <ListBox VerticalAlignment="Stretch" Background="black" x:Name="MasterDesLangages" MaxWidth="150" MinWidth="20" Padding="3,1"
71     SelectedItem="{Binding ItemsLangages.ItemCourant}" ItemsSource="{Binding ItemsLangages.LesItems}"
72     SelectionChanged="MasterDesLangages_SelectionChanged">
73     <ListBox.ItemTemplate>
74     <!--
75         Ce DataTemplate permet de contrôler l'affichage du
76         langage dans la ListBox et n'affiche que le nom du langage
77     -->
78     <DataTemplate>

```

1.15. Je sais gérer le DataBinding sur mon détail.

Cf. /source/repos/notre_solution/notre_projet/DescriptionLangageUC.xaml

1.16. Je sais gérer le DataBinding et les Dependency Property sur mes UserControl.

Cf. /source/repos/notre_solution/notre_projet/DescriptionLangageUC.xaml

1.17. Je sais développer un Master/Detail.

Cf. /source/repos/notre_solution/notre_projet/MainWindow.xaml

Cf. /source/repos/notre_solution/notre_projet/DescriptionLangageUC.xaml

2. Conception et Programmation Orientées Objets

2.1. Je sais concevoir un diagramme de classes qui représente mon application.

Cf. /documentation/csharp.pdf (page 1).

2.2. Je sais réaliser un diagramme de paquetages qui illustre bien l'isolation entre les parties de mon application.

Cf. /documentation/csharp.pdf (page 2).

2.3. Je sais décrire mes deux diagrammes en mettant en valeur et en justifiant les éléments essentiels.

Cf. /documentation/csharp.pdf (page 2, 3).

2.4. Je maîtrise les bases de la programmation C# (classes, structures, instances...).

Classes Cf./source/repos/notre_solution/notre_bibliotheque/Langage.cs

structures

Cf./source/repos/notre_solution/notre_bibliotheque/ValeurTri.cs

instances

Cf./source/repos/notre_solution/notre_projet/MainWindow.xaml.cs

2.5. Je sais utiliser l'abstraction à bon escient (héritage, interfaces, polymorphisme).

Cf. /source/repos/notre_solution/notre_projet/MainWindow.xaml.cs ligne 32

2.6. Je sais gérer des collections simples (tableaux, listes...).

Cf. /source/repos/notre_solution/notre_bibliotheque/Items.cs (propriété LesItems)

2.7. Je sais gérer des collections avancées (dictionnaires).

```

19     public Dictionary<string, IEnumerable<Item>> ChargerLesDonnées()
20     {
21         Dictionary<string, IEnumerable<Item>> DictionnaireÀRetourner = new Dictionary<string, IEnumerable<Item>>();
22         DonnésASerialiser tmp;
23         if (!File.Exists(FileNameCompleet))
24         {
25             throw new Exception("Le fichier spécifié n'existe pas");
26         }
27
28         DataContractSerializer serializer = new DataContractSerializer(typeof(DonnésASerialiser));
29         using(Stream fichierDeChargement = File.OpenRead(FileNameCompleet))
30         {
31             tmp = serializer.ReadObject(fichierDeChargement) as DonnésASerialiser;
32         }
33         DictionnaireÀRetourner.Add("Langages", tmp.LangagesASauvegarder);
34         DictionnaireÀRetourner.Add("Comptes", tmp.ComptesASauvegarder);
35         return DictionnaireÀRetourner;
36     }

```

/source/repos/notre_solution/PersDataContract/MyDataContract.cs

Cf. /source/repos/notre_solution/PersDataContract/MyDataContract.cs

2.8. Je sais contrôler l'encapsulation au sein de mon application.

Cf. /source/repos/notre_solution/notre_bibliotheque/Items.cs (attribut lesItems).

2.9. Je sais tester mon application.

Cf. /source/repos/notre_solution/test_fonctionnel/Program.cs

Cf. /source/repos/notre_solution/test_unitaire/*.cs

2.10. Je sais utiliser LINQ.

```

41     case ValeurTri.Date:
42         return lesItems.OrderBy(it => (it as Langage).DateDuLangage).ToList();
43     case ValeurTri.Favoris:
44         if (ItemsDeLAutreType.ItemCourant == null) return new List<Item>();
45         return (ItemsDeLAutreType.ItemCourant as Compte).LesLangagesFavoris.OrderBy(lang => (lang as Langage).Nom).ToList() as IList<Item>;
46     case ValeurTri.Génération:
47         return lesItems.OrderBy(it => (it as Langage).Generation).ToList();
48     case ValeurTri.Historique:
49         if (ItemsDeLAutreType.ItemCourant == null) return new List<Item>();
50         List<Item> historique = new List<Item>();
51         foreach(Item lang in (ItemsDeLAutreType.ItemCourant as Compte).HistoriqueDuCompte)
52         {
53             historique.Add(lang);
54         }
55         historique.Reverse();
56         return historique;
57     default:
58         return lesItems.OrderBy(it => (it as Langage).Nom).ToList();

```

/source/repos/notre_solution/notre_bibliotheque/Items.cs ligne 41~58

2.11. Je sais gérer les évènements.

```

15     public class Items : IEquatable<Items>, INotifyPropertyChanged
16     {
17         private ValeurTri filtre;
18         public ValeurTri Filtre
19         {
20             get => filtre;
21             set
22             {
23                 filtre = value;
24                 OnPropertyChanged("LesItems");
25             }
26         }
27
28         public Item ItemCourant { get; set; }
29         public Items ItemsDeLAutreType { get; set; }
30         private IList<Item> lesItems;
31
32         public event PropertyChangedEventHandler PropertyChanged;

```

Figure 1:

/source/repos/notre_solution/notre_bibliotheque/Items.cs (lignes 15, 24, 32)

3. Projet Tuteuré S2

3.1. Je sais mettre en avant dans mon diagramme de classes la persistance de mon application.

Cf. /documentation/projet_tutoré.pdf (page 3)

3.2. Je sais mettre en avant dans mon diagramme de classes ma partie personnelle.

Cf. /documentation/projet_tutoré.pdf (page 3)

3.3. Je sais mettre en avant dans mon diagramme de paquetages la persistance de mon application.

Cf. /documentation/projet_tutoré.pdf (page 3)

3.4. Je sais réaliser une vidéo de 1 à 3 minutes qui montre la démo de mon application.

Cf. /documentation/Presentation.flv

3.5. Je sais coder la persistance au sein de mon application.

Cf. /source/repos/notre_solution/Données/Stub.cs

Cf. /source/repos/notre_solution/PersDataContract/MyDataContract.cs

Cf. /source/repos/notre_solution/notre_bibliotheque/IPersistence.Cs

3.6. Je sais coder une fonctionnalité qui m'est personnelle.

```

38     private void ConnexionAUnCompte_Click(object sender, RoutedEventArgs e)
39     {
40         GestionnaireDeComptes g = ((App.Current as App).MainWindow as MainWindow).GestionnaireCompte;
41         if (g.IsSomeoneConnected)
42         {
43             MessageBox.Show("Vous êtes déjà connecté", "connexion", MessageBoxButton.OK, MessageBoxImage.Error);
44         }
45         else
46         {
47             if (String.IsNullOrEmpty(idDuCompteSaisi.Text.Trim()) || String.IsNullOrEmpty(mdpSaisi.Password.Trim()))
48             {
49                 MessageBox.Show("Vous devez remplir les champs", "connexion", MessageBoxButton.OK, MessageBoxImage.Error);
50             }
51             else
52             {
53                 Compte tmp = new Compte(idDuCompteSaisi.Text, mdpSaisi.Password);
54                 MenuItem boutonLangage = (App.Current.MainWindow as MainWindow).boutonLangages;
55                 if (g.ItemsComptes.LesItems.Contains(tmp))
56                 {
57                     int indexCompte = g.ItemsComptes.LesItems.IndexOf(tmp);
58                     g.ItemsComptes.ItemCourant = g.ItemsComptes.LesItems[indexCompte] as Compte;
59                     (App.Current.MainWindow as MainWindow).temoinID.DataContext = g.ItemsComptes.ItemCourant;
60                     boutonLangage.DataContext = g.ItemsComptes.ItemCourant;
61                 }
62                 else
63                 {
64                     MessageBox.Show("L'identifiant ou le mot de passe est incorrecte", "connexion", MessageBoxButton.OK, MessageBoxImage.Error);
65                 }
66             }
67         }
68     }
69 }

```

/source/repos/notre_solution/notre_projet/ConnexionUC.xaml.cs connexion à un compte

3.7. Je sais documenter mon code.

Cf./source/repos/notre_solution/Notre_bibliotheque/Items.cs

3.8. Je sais utiliser SVN.

3.9. Je sais développer une application qui compile.

Cf. /source/repos/notre_solution/ProLangBin/netcoreapp3.0/vues.exe

3.10. Je sais développer une application fonctionnelle.

Cf./documentation/Presentation.flv

3.11. Je sais mettre à disposition un outil pour déployer mon application.

Publication :

Cf./source/repos/notre_solution/ProLangBin/netcoreapp3.0/publish

Installation :

Cf./source/repos/notre_solution/ProLang_SetUp

Cf./source/repos/notre_solution/ProLangBin/setup/ProLang_SetUp.msi