

# PREUVES

## **Théo SOUCHON & Théophile DECAESTEKER**

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

Voir la rubrique "Contexte" de la documentation (page 1 à 2).

### **Je sais dessiner des sketches pour concevoir les fenêtres de mon application**

Voir la rubrique "Sketchs" de la documentation (page 3 à 4).

### **Je sais enchaîner mes sketches au sein d'un story-board**

Voir la rubrique "Story-board" de la documentation (page 6 à 10).

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

Voir la rubrique "Diagramme d'utilisation" & "Description du diagramme d'utilisation" de la documentation (page 1 à 17).

### **Je sais concevoir un diagramme de classes qui représente mon application**

Voir diagramme de classe dans la documentation.

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

Voir diagramme de paquetage dans la documentation.

## **Je sais décrire mes deux diagrammes en mettant en valeur et en justifiant les éléments essentiels**

Voir description du diagramme de paquetage et de classe.

## **Je sais concevoir une application ergonomique**

Utilisation de balise xaml : <grid> avec les propriétés MaxHeight et MaxWidth.

(Présent dans chaque fenêtre xaml)

Utilisation de la propriété WindowStartupLocation="CenterScreen" de <Window> afin de faire apparaître toutes les fenêtres au centre de l'écran.

Utilisation de couleur en tant que background : #ECE1DE et de styles dans app.xaml afin de créer des boutons, textboxes et textblocks aux coins arrondis sur fond blanc et bord noir.

Éléments de la vue suffisamment espacés grâce aux "grid" et aux "margin".

## **Je sais concevoir une application avec une prise en compte de l'accessibilité**

Textes et titres sans fautes d'orthographe et écrit en français.

Utilisation de Titre et zones de textes afin de guider l'utilisateur sur les pages.

Reprise des codes traditionnels d'une application afin de permettre à l'utilisateur d'utiliser l'application de manière innée. (ex: Boutons connexions et inscriptions en haut de la page, Titre en haut au milieu avec barre de recherche).

## **Je sais mettre en avant dans mon diagramme de classes la persistance de mon application**

Voir diagramme de classe dans la documentation.

## **Je sais mettre en avant dans mon diagramme de classes ma partie personnelle**

Les fonctionnalités supplémentaires ajoutées et non obligatoire sont le bouton 'Dark Mode' de l'application (Mode nuit) et la fonction de recherche dans Accueil.xaml.cs. J'ai aussi été amené à apprendre à utiliser "ObjectDataProvider" (voir app.xaml).

## **Je sais mettre en avant dans mon diagramme de paquetages la persistance de mon application**

Voir diagramme de paquetage dans la documentation.

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

L'intégralité du projet fait acte de preuve sur ma capacité à avoir maîtrisé les bases de la programmation en C#.

## **Je sais utiliser l'abstraction à bon escient (héritage, interfaces, polymorphisme)**

On utilise une interface afin d'utiliser les fonctions chargement et sauvegarde dans le cadre de la persistance de l'application. (voir classe 'Persistance' dans la paquet : CsharpCode.)  
On utilise aussi l'héritage dans chaque classes ou 'code behind' de fenêtre sous la forme de using. Que ce soit pour importer des classes personnelles ou des bibliothèques déjà présentes dans Visual Code.

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

On utilise des "List< >" et des "ObservableCollection" et on sait différencier l'utilisation des deux collections. (voir CsharpCode et les classes commençant par 'Liste...' afin d'y trouver "ObservableCollection".

## **Je sais gérer des collections avancées (dictionnaires).**

Nous n'avons pas utilisé de dictionnaires.

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

Nous n'avons pas utilisé d'encapsulation, en effet nos listes sont modifiable, affichable, lisible ne nous permettant donc pas de faire une encapsulation pour un type d'action précis.

## **Je sais tester mon application. Je sais utiliser LINQ.**

Nous avons utilisé un projet de type console afin d'afficher des données depuis le stub pour tester notre application au début. Nous n'avons pas utilisé LINQ.

## **Je sais gérer les évènements.**

Nous avons utilisé des évènements d'éléments xaml sur chaque fenêtre xaml tout au long du projet. Voici quelques exemples qui nous semblent importants : 'Click', 'TextChanged', 'SelectionChanged', 'PreviewTextInput', 'Closing'.

## **Je sais choisir mes layouts à bon escient**

Voir vue de l'application.

## **Je sais choisir mes composants à bon escient.**

Voir les éléments du xaml des différentes fenêtres du projet.

## **Je sais créer mon propre composant.**

Voir les User-controls dans le projet SugarDay, dans le répertoire /UC/.

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

Voir App.xaml.cs dans les balises <Styles> (On les utilise sur tous les boutons, textblocks et boxs de la vue).

Voir ObjectDataProvider dans App.xaml.cs (utilisé pour la comboBox dans AjouterIngredient.xaml)

## **Je sais utiliser les DataTemplate (locaux et globaux).**

DataTemplate locaux : voir chaque listBox / ComboBox (ex: Accueil.xaml)

## **Je sais intercepter les événements de la vue.**

Utilisation des événements 'Click' , 'SelectionChanged' , 'TextInputPreview'.

## **Je sais notifier la vue depuis des événements métier.**

Utilisation des textBlocks qui affiche du texte après un événement. (ex: Après événement 'Click' voir les textblocks ErrorConnexion et ErrorInscription dans Connexion et Inscription).

## **Je sais gérer le DataBinding sur mon master.**

Voir binding des 3 ListBoxes dans *Accueil.xaml.cs*.

## **Je sais gérer le DataBinding sur mon détail.**

Voir Binding des 2 ListBoxes dans *DétailsDeRecette.xaml.cs*.

Voir Binding des 2 ListBoxes dans *FenetreIngredientActuel.xaml.cs* et *FenetreUstensileActuel.xaml.cs*.

Voir Btn\_Sauvegarde() dans *Profil.xaml.cs* .

## **Je sais gérer le Data Binding et les Dependencyproperty sur mes UserControl.**

Voir les 2 ContentControls dans *accueil.xaml.cs* et les 4 user-controls (*Deconnexion\_button.xaml.cs* | *Non\_connecté.xaml.cs* | *ProfilButton.xaml.cs* | *UConnexion.xaml.cs*) dans le répertoire /UC/ dans SugarDay .

## **Je sais développer un Master/Detail.**

Voir le code de l'application.

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

Voir dossier 'Vidéo' sur la forge.

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

voir /SVN/Code → classe **Persistence.cs** : pour la création.

voir /SVN/Code → classe *ListeCommentaire.cs* | *ListeIngredient.cs* | *ListeUtilisateur.cs* : pour l'instruction des méthodes **chargement()** et **sauvegarde()**.

voir /SVN/Code → **Stub.cs** pour les appels

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

Les fonctionnalités personnelles sont le Mode nuit (DarkMode) de l'application (voir /SVN/Code → classe *Accueil.xaml.cs* → méthode *DarkMode()* ) et la barre de recherche (voir /SVN/Code → classe *Accueil.xaml.cs* → méthode *Barre\_de\_recherche()* ) .

## Je sais documenter mon code.

Voir répertoire 'Documentation' sur SVN.

## Je sais utiliser SVN.

L'ensemble des commits présent sur la forge avec les commentaires associés. Et l'architecture des fichiers dans le dépôt de SVN qui est conforme à ce qui est attendu. (voir dépôt "DECAESTEKER - SOUCHON" → *csharp\_applicuisine*)

## Je sais développer une application qui compile.

Rendu de l'application finale sans erreur de code ou de conflit.

## Je sais développer une application fonctionnelle.

Voir l'exécutable de l'application.

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

exécutable disponible a cet endroit :

[https://drive.google.com/file/d/1L8q\\_tE2L9ed\\_cMNmrq43dN9F4S\\_iryZi/view?usp=sharing](https://drive.google.com/file/d/1L8q_tE2L9ed_cMNmrq43dN9F4S_iryZi/view?usp=sharing)

