

## ITEM FINDER

### I/ Présentation de l'application

Bonjour à tous, l'application que nous avons créée s'appelle *ItemFinder*. Le projet consiste en une **application de filtrage d'annonce** en ligne qui permet aux utilisateurs de trouver plus **rapidement** et plus facilement les annonces qui les intéressent. Le but est de proposer un service complémentaire aux sites d'annonces existants en proposant des **filtres plus avancés** ou même des **alertes en temps réel** pour permettre à l'utilisateur de pouvoir profiter des meilleures affaires. Pour l'instant notre application est basée sur le site *LeBonCoin* et principalement sur le domaine du **multimédia** mais l'avantage est que le principe est assez facilement adaptable pour correspondre à différents sites d'annonces et dans différents domaines

Tout d'abord il y a un **menu de recherche**, permettant de taper les mots clés pour la recherche comme sur un site d'annonce classique avec le modèle d'un téléphone par exemple l'iPhone 13.

Ensuite nous proposons un large menu de filtres qui permettent à l'utilisateur d'effectuer une **recherche plus ciblée** et d'obtenir des résultats **pertinents** sans avoir à scroller pendant plusieurs dizaines de minutes avant de trouver le produit souhaité et à un prix intéressant. Par exemple sur *LeBonCoin* il n'est pas rare en tapant dans la barre de recherche iPhone 13, de tomber sur des coques de téléphone ou encore seulement certaines pièces du téléphone comme l'écran par exemple.

A l'origine du projet, le but était également de proposer un **système d'alertes** envoyées directement sur le téléphone de l'utilisateur via une des applications existantes comme *IFTTT* par exemple. Malheureusement nous avons peut-être été un peu **trop ambitieux** et nous n'avons pas eu le temps de développer cette fonctionnalité que nous ajouterons sûrement plus tard pour un usage personnel de notre application. Cette fonctionnalité pourrait être un gros **point positif** pour notre application car **aucune** plateforme ne propose encore un service réactif de ce type et cela pourra donc permettre à l'utilisateur de se positionner sur une vente beaucoup plus rapidement que les autres et sans avoir à passer son temps à actualiser les résultats du site.

Enfin, notre application propose également une **redirection** sur *LeBonCoin*, qui permet aux utilisateurs de **visualiser rapidement** les annonces correspondantes sur le site d'origine. Cela facilite grandement le **processus d'achat**, en évitant aux utilisateurs de devoir rechercher l'annonce sur le site *LeBonCoin* après avoir trouvé une annonce intéressante sur notre application.

## II/ Présentation de l'application

L'interface de notre application a été conçue pour être **simple** et **intuitive**. Nous avons choisi de ne pas la rendre **redimensionnable** afin de garantir une expérience utilisateur **ergonomique** et qui conserve le design souhaité. Nous avons opté pour un design épuré et minimaliste, avec une palette de couleurs pales qui rappellent l'aspect du **papier**.

Chaque fenêtre de l'application a été conçue de manière à présenter clairement les fonctionnalités principales de l'application.

La première page d'accueil est la **page centrale** de notre application, elle permet d'indiquer la majorité des critères de recherches et les mots clés. Tout comme une **recherche leboncoin classique**, on peut sélectionner un ou des états des produits souhaités, la localisation et le budget. L'intérêt est que l'utilisateur peut disposer de **filtres supplémentaires non disponibles sur leboncoin**, comme d'afficher uniquement les annonces avec expédition ou encore exclure les annonces marquées comme étant en cours de vente avec le système de paiement et d'expédition sécurisé mis en place par la plateforme du *boncoin*. Il est aussi possible de consulter les annonces de vendeurs ayant une **bonne réputation**, grâce aux notes attribuées par les membres entre 0 et 5 étoiles. Il est ainsi possible d'exclure les profils **peu recommandés** ou ne disposant pas encore de notes (comme par exemple pour un membre récemment inscrit). Cela permet de limiter les risques **d'arnaque** lors des transactions. Par ailleurs, l'une des fonctionnalités les plus utiles et paramétrable sur cette page est le **filtre de pertinence**. En effet, ce dernier permet de calculer automatiquement **la moyenne des prix** des annonces, de manière séparée pour chacun des états. En effet, faire **une seule moyenne** à partir d'une liste d'annonces comportant des produits comme neuf et pour pièces par exemple et donc avec une différence de prix élevé n'est pas significative. Il est bien sûr possible de renseigner une **moyenne personnalisée** mais dans ce cas elle sera utilisée pour tous les états. Une fois cette moyenne établie, un autre filtre peut être appliqué, qui est l'**écart avec la moyenne**. Par exemple pour un produit avec une moyenne de 100 euros si on met un écart bas de 10% et un écart haut de 20%, les annonces affichées seront entre 90 et 120 euros. Par ailleurs, plus le prix est proche de la moyenne, et plus le pourcentage de **pertinence** sera élevé.

Lorsque l'on clique sur le bouton **rechercher**, l'utilisateur est redirigé vers une seconde page qui permet d'afficher les **résultats**. Nous pouvons classer par **pertinence** grâce à notre filtre décrit auparavant, ou simplement par **nouveautés**. Toutes les informations nécessaires pour déterminer si l'annonce est intéressante sont disponibles. **La moyenne automatique** calculée par le programme est aussi affichée pour guider l'utilisateur sur la valeur du produit sur le marché.

Une troisième page permet d'afficher les **recherches sauvegardées** qui sont configurées depuis une case à cocher sur la première page. Un système **d'alerte** avec fréquence de vérification **intelligente** aurait dû être mis en place et configurable sur cette page.

Une dernière fonctionnalité permet de mieux affiner les résultats grâce à un système de **liste noire**. Il est possible de renseigner plusieurs termes, et les annonces comportant dans leurs titres les termes définis seront exclues. Par exemple, cela peut être pratique pour des modèles de téléphone où il existe souvent plusieurs variantes pour une même gamme. Si je cherche un Huawei P30 Pro, je pourrais aussi voir apparaître des Huawei P30 Lite par exemple, alors que cela ne m'intéresse pas. Je peux ainsi **éliminer** plus précisément les produits.

### III/ Fonctionnement interne du programme

Tout d'abord nous avons vraiment fait en sorte que le code soit très modulable par le biais de classes et d'énumérations pour pouvoir le mettre à jour facilement et qu'il soit facilement compréhensible étant donné l'importance du projet. Par exemple il est tout à fait possible de copier l'entièreté du code qui permet de récupérer des annonces sur LeBonCoin et de l'intégrer à un nouveau projet en quelques lignes. Nous avons aussi créé nos propres composants avec leur logique interne, donc il est aussi possible de les copier facilement en un simple clic vers un autre projet pour les intégrer à son interface.

Maintenant pour vous donner une idée, voici comment l'application fonctionne chronologiquement :

- Les entrées utilisateurs sont vérifiées : c'est-à-dire nous faisons en sorte de vérifier que le champ de recherche est bien rempli, le code postal est bien un chiffre, etc. et le cas échéant nous affichons un message d'erreur
- Les paramètres de recherche sont enregistrés sous forme de classe et transmis à un code pour récupérer les annonces sur LeBonCoin
- Ensuite nous appliquons les filtres sur les annonces récupérés à l'aide d'une classe qui prend en compte les paramètres fournis par l'utilisateur
- Nous ajoutons les paramètres de la recherche actuelle à un fichier de sauvegarde existant si l'utilisateur a choisi d'enregistrer la recherche
- Puis nous pouvons enfin afficher ces résultats dans une liste.

On va maintenant rentrer dans les détails des étapes importantes notamment la phase de récupération des annonces à partir de LeBonCoin, et celle qui permet de filtrer ces annonces.

Alors pour récupérer des annonces à partir de LeBonCoin :

- Nous envoyons tout d'abord une requête à la page de recherche de LeBonCoin avec le texte de la recherche en paramètre
- Puis nous récupérons le code HTML de cette page et faisons une sélection sur les éléments HTML qui nous intéressent : le nom, le prix, la note du vendeur, la localisation, l'image

**Ces 2 étapes sont appelées le scraping**

- Ensuite nous convertissons ces données à la base toutes en chaînes de caractères en type valide. Le prix est converti en entier, la date en type date, etc. Cette étape est très utile pour appliquer les filtres sur les annonces par la suite : par exemple pour classer les annonces par prix croissant, ou imposer un minimum sur la note du vendeur il faut que ces paramètres soient des entiers et non des chaînes de caractères.

**Cette étape est appelée le parsing**

- Les annonces sont ensuite disponibles sous forme d'une liste d'objet Annonce, cet objet contient les propriétés d'une annonce : prix, nom, date, localisation, nom du vendeur, etc.

Après avoir récupéré les annonces, il faut les filtrer sans quoi notre projet serait inutile, pour cela nous procédons de cette manière :

- A partir de la liste d'annonce globale, nous allons sélectionner les annonces correspondant à chaque état de produit spécifié par l'utilisateur pour les ajouter à des listes bien distinctes en fonction de leur état. Cela va permettre de calculer une moyenne de prix pour chaque liste et donc pour chaque état.
- Ensuite dans chaque liste nous sélectionnons les annonces dont le prix est dans l'intervalle de moyenne (indiqué en pourcentage) spécifié par l'utilisateur, en fonction de la moyenne des prix de l'état.

## Documentation

- On ajoute ensuite une propriété de pertinence à chaque annonce, plus ce score est élevé plus le prix de l'annonce est proche de la moyenne correspondant à son état.
- Puis on va combiner toutes ces listes en une seule liste
- Ensuite on va appliquer tous les autres filtres sur cette liste globale comme le filtre imposant un minimum de note pour le vendeur, la livraison disponible, les annonces à exclure comportant certains mots, etc.

*A propos*

### IV/ Webographie, source et organisation

Pour ce qui est de notre organisation, nous nous sommes régulièrement retrouvés soit chez nous, soit à la bibliothèque universitaire pour travailler tous ensemble, réfléchir aux différentes fonctionnalités ou corriger les différents problèmes rencontrés.

Nous nous sommes aussi séparé les tâches, avec certaines personnes qui avançait plutôt sur le code interne pendant que d'autres avançait plutôt sur la partie graphique et le développement de l'interface. Nous avons utilisé un groupe WhatsApp pour communiquer et un google drive pour les fichiers à échanger.

Pour créer le **design de l'interface** avant de la coder nous avons utilisé *Figma*, un logiciel de design graphique gratuit qui a pour avantage d'être accessible par tous les membres du groupe en même temps.

Nous avons rencontré plusieurs difficultés pour la création de notre projet, mais la principale a clairement été le manque de temps pour développer le grand nombre de fonctionnalités que nous voulions ajouter. En effet lors de la création du projet nous avons voulu faire une application qui en plus de nous intéresser nous sera utile par la suite et donc nous avons voulu ajouter le plus d'options possibles. Pour l'organisation nous n'avons pas assez pris en compte la très grosse charge de travail personnelle liée aux nombreux contrôles du semestre et donc nous avons eu moins de moment de temps libre pour avancer le projet que prévu.

Ensuite certaines fonctionnalités comme le scraper du site *LeBonCoin* étaient également compliquées à développer car le site possède des **anti-robots** difficiles à contourner. Nous avons aussi fait le choix de ne pas utiliser de *template* préexistante pour le design de l'interface mais de la créer entièrement nous-même donc cela a pris beaucoup de temps pour avoir un rendu qui nous plaisait.

Comme sources nous avons utilisé la documentation C# de *Microsoft* pour apprendre les bases du langage et également certains forums comme *StackOverflow* pour la résolution de problèmes de code.