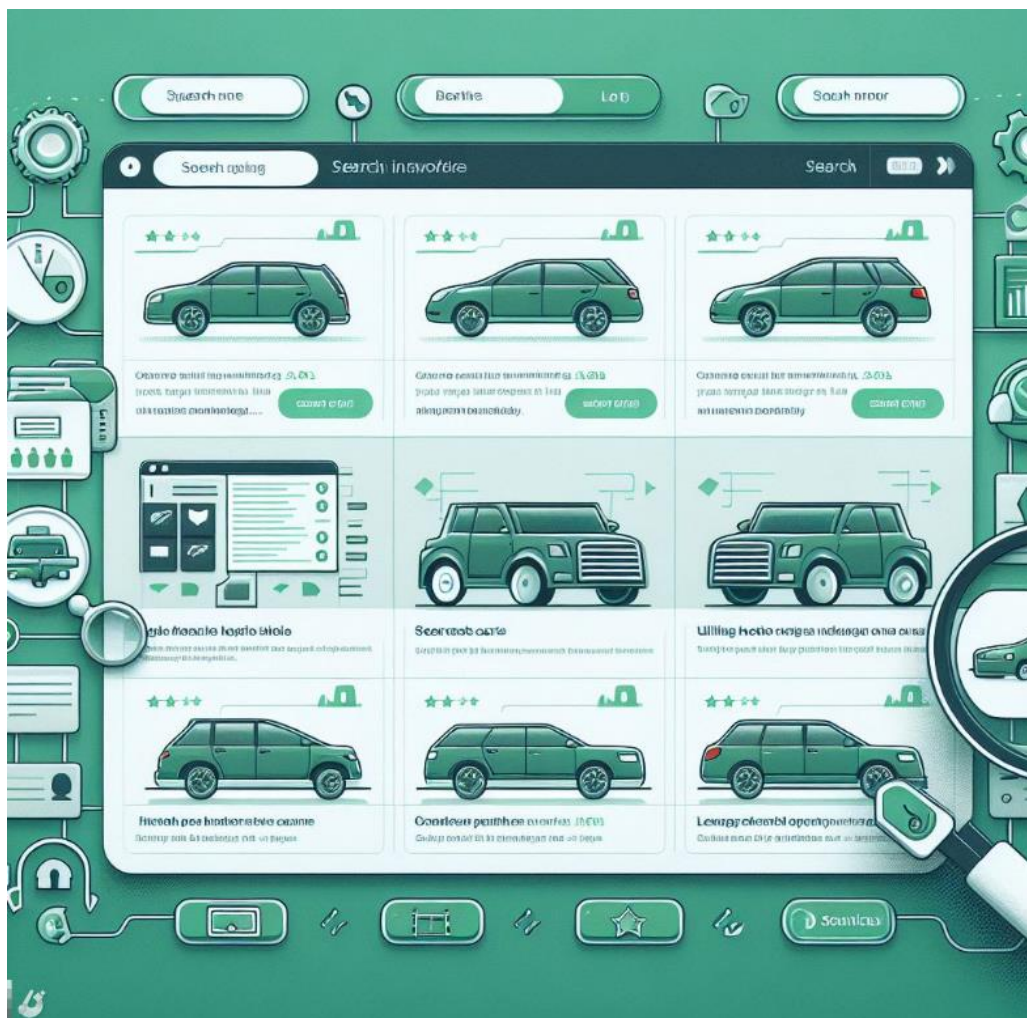


Application C# de type plateforme de vente de voitures



Maikol Correia Da Silva
Maikol.Correia@eduvaud.ch

SI-CA2a

30.04.2024 - 29.05.2024

Table des matières

1	Introduction.....	4
1.1	Cadre, description et motivation.....	4
1.2	Organisation	4
1.3	Objectifs.....	4
1.4	Planification initiale	5
2	Analyse.....	7
2.1	Cahier des charges détaillé.....	7
2.1.1	MCD.....	7
2.1.2	Maquettes	8
2.1.3	Use Cases et scénarios	20
2.2	Stratégie de test	21
2.3	Planification	22
2.3.1	Sprint 1	22
2.3.2	Sprint 2	22
2.3.3	Sprint 3	23
2.3.4	Sprint 4	23
3	Conception.....	24
3.1	Logiciels.....	24
3.2	Choix techniques	24
3.3	MLD.....	25
3.4	Diagramme de classes	26
3.5	Diagramme de flux	27
4	Réalisation	28
4.1	Dossier de réalisation	28
4.1.1	Sprint 1	28
4.1.2	Sprint 2	29
4.1.3	Sprint 3	33
4.1.4	Sprint 4	34
4.2	Description des tests effectués	35
4.3	Test de l'Application par un Utilisateur Réel	38
4.4	Erreurs restantes	38
5	Conclusion	39
6	Annexes	40
6.1	Résumé du rapport du TPI.....	40
6.2	Glossaire	41
6.3	Sources – Bibliographie	42
6.4	Journal de Travail.....	44
6.5	Planification finale	50

1 Introduction

1.1 Cadre, description et motivation

Ce travail est réalisé dans le cadre du TPI durant ma seconde année de FPA au CPNV, d'avril 2024 à mai 2024. Ce projet vise à créer une application simple d'utilisation qui permet la vente et l'achat de voitures d'occasion, tout en mettant l'accent sur la gestion efficace d'une base de données. En tant que passionné de voitures, je suis enthousiaste à l'idée de pouvoir développer un logiciel qui porte sur ce sujet.

1.2 Organisation

Élève 1 : Correia Da Silva, Maikol, maikol.correia@eduvaud.ch

Responsable de projet : Benzonana, Pascal, pascal.benzonana@eduvaud.ch

Expert 1 : Montemayor, Ernesto, ernesto@bati-technologie.ch

Expert 2 : Melly, Jonathan, Jonathan.melly@eduvaud.ch

Un rendu semi-hebdomadaire sera effectué tous les mercredis et vendredis. Le rendu se fera par mail et contiendra les différents liens utiles tels que GitHub et IceScrum ainsi que les PDF de la documentation et du journal de travail.

1.3 Objectifs

L'objectif principal de ce projet est de créer une application en C# permettant de vendre et acheter des voitures. Les données seront centralisées dans une base de données et les échanges entre acheteurs et vendeurs se feront via cette dernière.

L'application sera composée de 3 parties : administrateur, vendeur/acheteur, publique.

Dans la partie publique, l'utilisateur pourra :

- Voir les véhicules mis en vente
- Rechercher des véhicules selon différents filtres tels que :
 - Marque
 - Modèle
 - Année
 - Prix
 - Type de motorisation

Dans la partie vente/achat, après s'être connecté ou inscrit, l'utilisateur pourra :

- Procéder aux mêmes opérations que dans la partie publique
- Mettre une annonce en favori
- Mettre en vente un véhicule en insérant ses données via un formulaire
- Modifier ou supprimer ses annonces (sauf si le véhicule a été vendu)
- Acheter un véhicule

Dans la partie administration, l'utilisateur pourra :

- Faire la même chose que dans la partie vente/achat
- Bloquer un utilisateur
- Bloquer une annonce

1.4 Planification initiale

Le projet a débuté le 30 avril 2024 à 8h00 et se termine le 29 mai 2024 à 11h35 avec un total de 90 heures. La répartition du travail pour ce projet est la suivante : l'analyse représente 20%, l'implémentation 45%, les tests 10% et la documentation 25%.

Pour ce projet, j'utilise la méthode agile car elle me permet une meilleure flexibilité au niveau du travail et car je suis habitué à utiliser cette méthode dans mes projets. Au total, le projet contient quatre sprints avec des longueurs différentes afin que chaque sprint se termine lorsque je vois mon chef de projet, ainsi nous pouvons faire un sprint review qui me permettra de passer au sprint suivant.

Projet

Application C# de type plateforme de vente de voiture

Planification

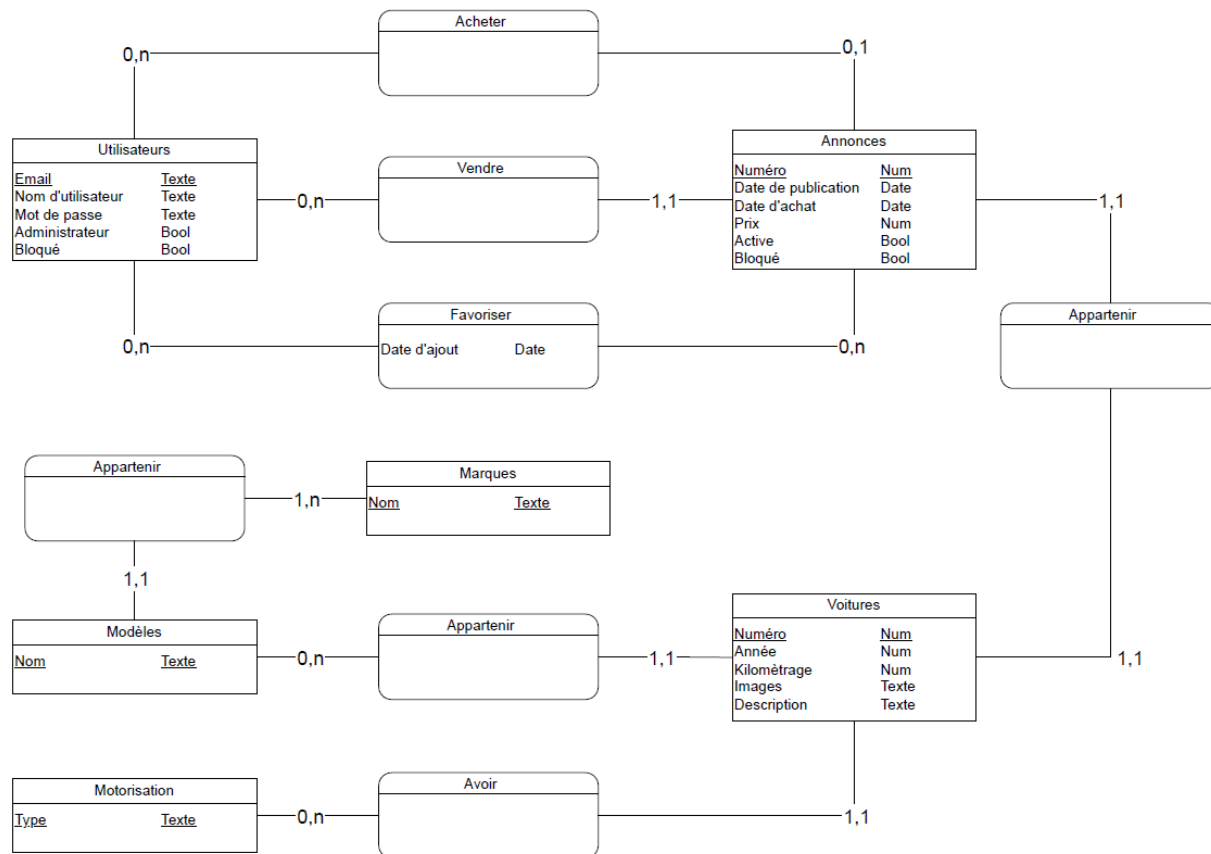
Total		30.04.24	07.05.24	17.05.24	24.05.24	
Prévu	90 h 00	25 h 25	28 h 40	19 h 05	16 h 50	
Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	0 h 00	
		Sprint 1	2	3	4	
1 Analyse						18 h 00
						0 h 00
1.1 - Lecture du cahier des charges	Prévu	0 h 30				0 h 30
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu		0 h 40	0 h 40		1 h 20
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu	3 h 25				3 h 25
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu	4 h 40				4 h 40
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu	3 h 30				3 h 30
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu	3 h 30				3 h 30
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu					
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu	0 h 20	0 h 25	0 h 20		1 h 05
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
2 Implémentation						40 h 30
						0 h 00
2.1 - Mise en place de l'environnement	Prévu	3 h 30				3 h 30
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu	3 h 00				3 h 00
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu		4 h 00			4 h 00
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu		5 h 00			5 h 00
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu		4 h 00			4 h 00
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu		4 h 00			4 h 00
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
2.2 - Base de données	Prévu		2 h 00			2 h 00
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu			2 h 30		2 h 30
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu			3 h 00		3 h 00
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu			4 h 30		4 h 30
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu			2 h 30		2 h 30
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu			2 h 30		2 h 30
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
3 Tests						9 h 00
						0 h 00
3.1 - Tests unitaires	Prévu		1 h 00	1 h 15	0 h 30	2 h 45
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu		1 h 35	2 h 00	1 h 40	5 h 15
3.2 - Tests d'intégrations	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
	Prévu				1 h 00	1 h 00
3.3 - Test d'acceptation	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
4 Documentation						22 h 30
						0 h 00
41 - Journal de travail	Prévu	2 h 00	1 h 00	0 h 40	0 h 40	4 h 20
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	
42 - Rédaction de la documentation	Prévu	1 h 00	5 h 00	4 h 10	8 h 00	18 h 10
	Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	

Figure 1 : Planification Initiale

2 Analyse

2.1 Cahier des charges détaillé

2.1.1 MCD



Projet	TPI Vente de voiture d'occasion
Titre	SellBuyAuto
Auteur	Maikol Correia Da Silva
Version	v1.0 - 8 mai 2024

Figure 2 : MCD

Dans ce Modèle Conceptuel de Données (MCD), j'ai délibérément séparé les entités « Annonces » et « Voitures ». Cette séparation vise à améliorer la lisibilité de la base de données mais également à prévoir une évolution du programme. En effet, cette conception permet d'ajouter facilement d'autres types de véhicules, tels que des motos, sans affecter la structure existante de la base de données.

2.1.2 Maquettes

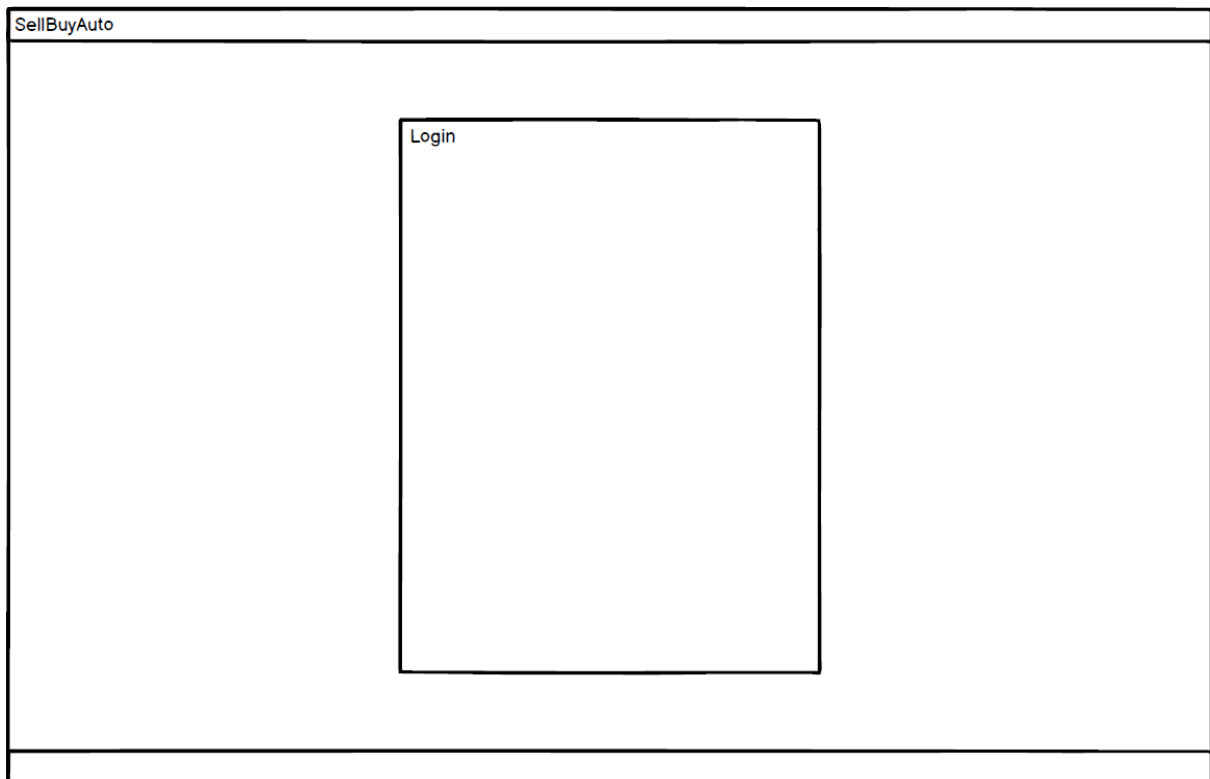


Figure 3 : LoginPage_Zonning

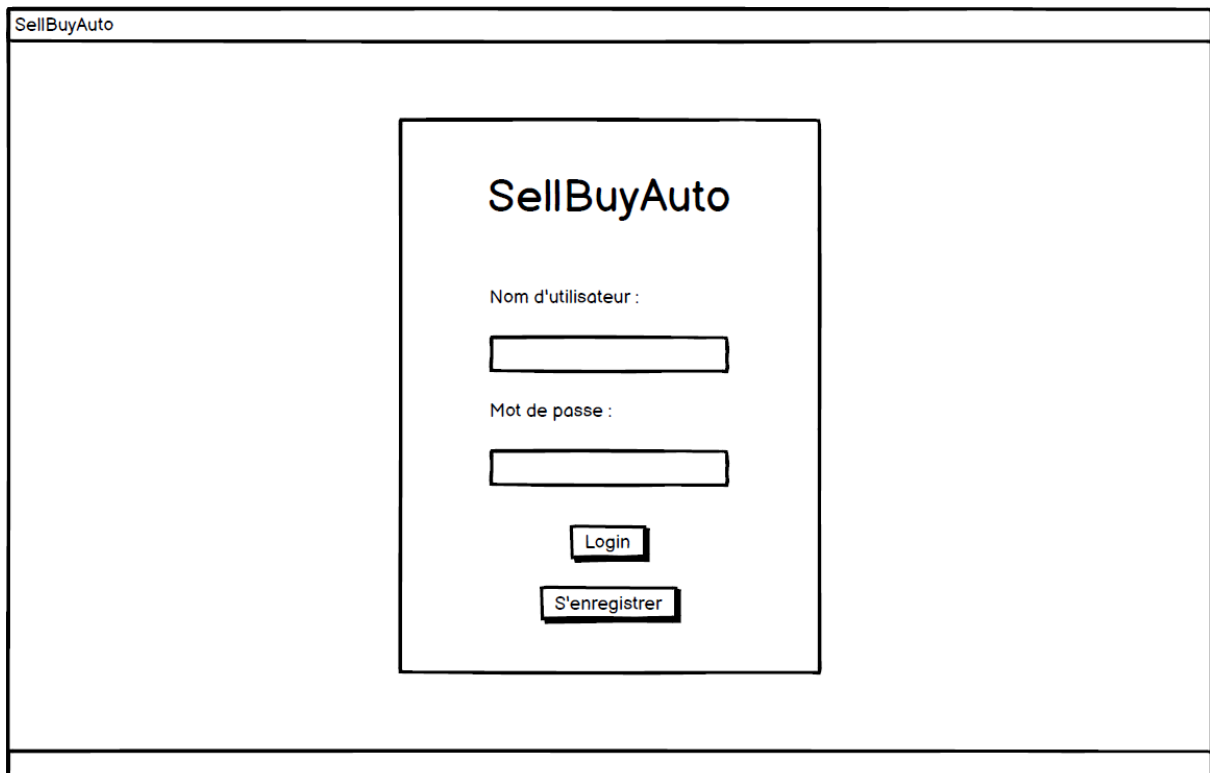


Figure 4 : LoginPage_Wireframe

Cette page correspond à la page de connexion. Elle permettra donc à l'utilisateur de se connecter à son compte.

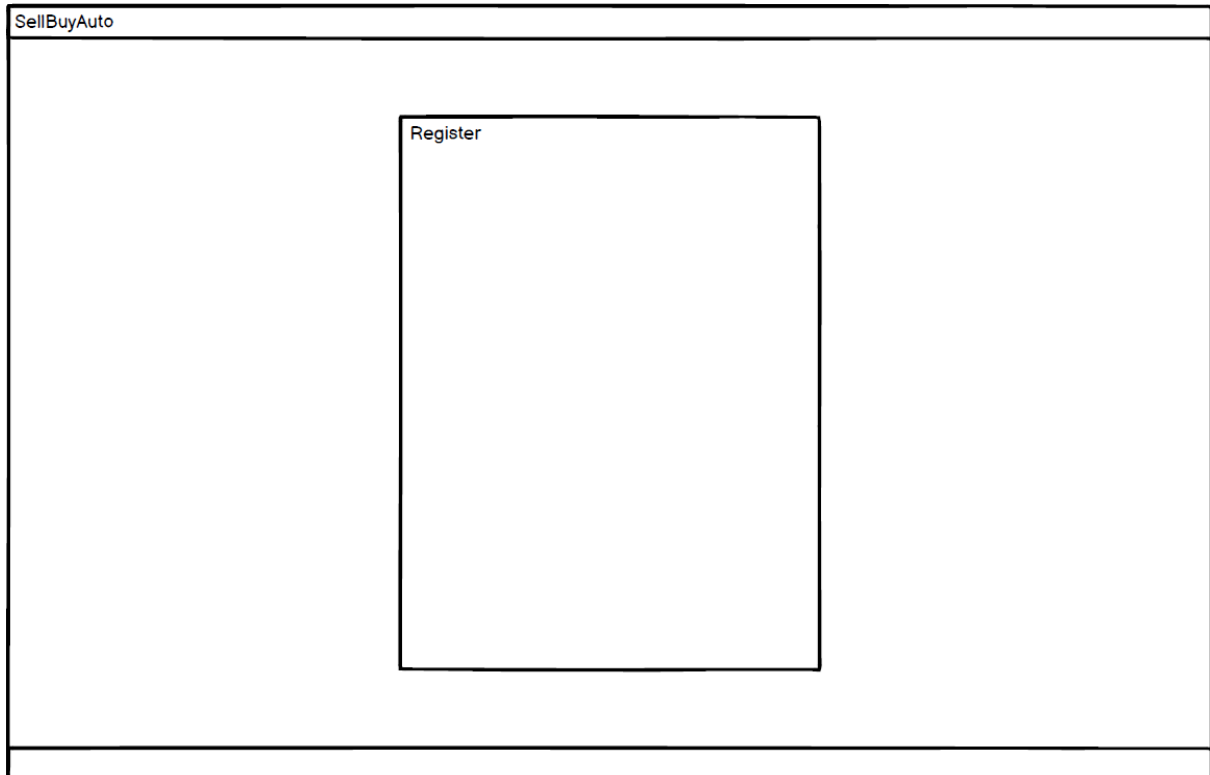


Figure 5 : RegisterPage_Zonning



Figure 6 : RegisterPage_Wireframe

Cette page correspond à la page d'enregistrement. Elle permettra donc à l'utilisateur de se créer un compte.

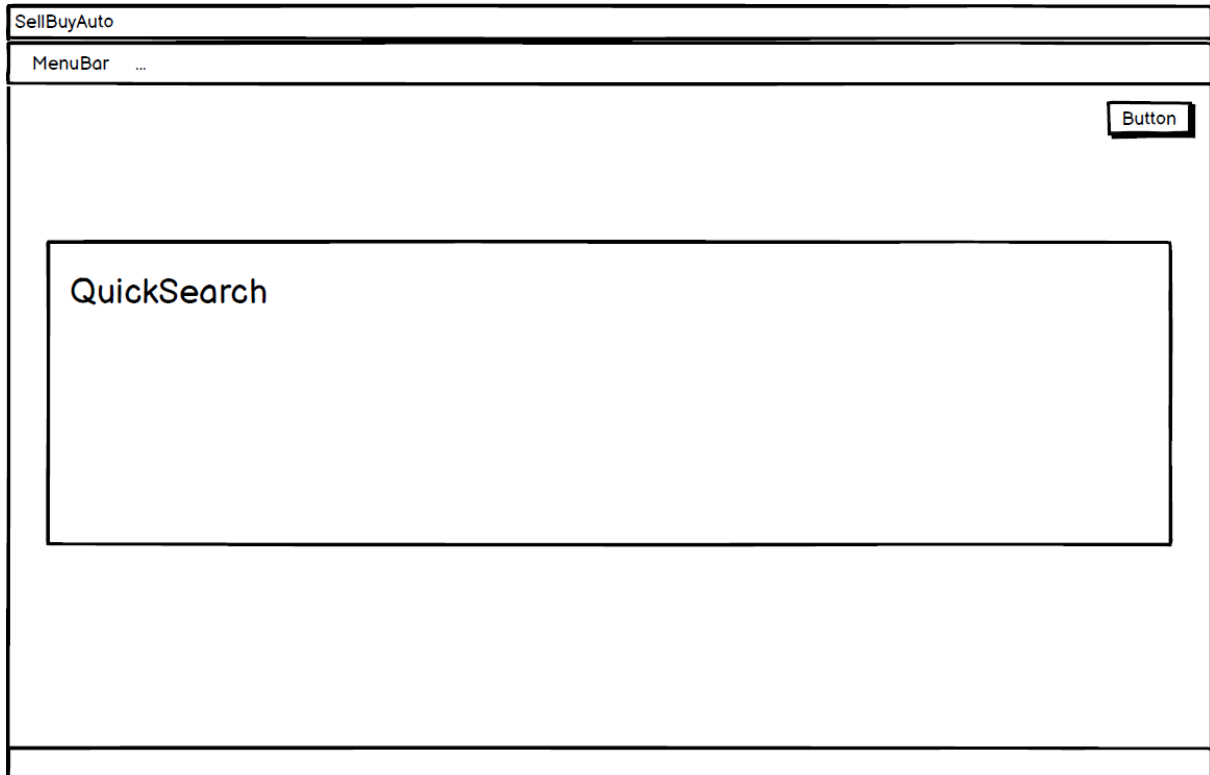


Figure 7 : HomePageWithoutLogin_Zonning

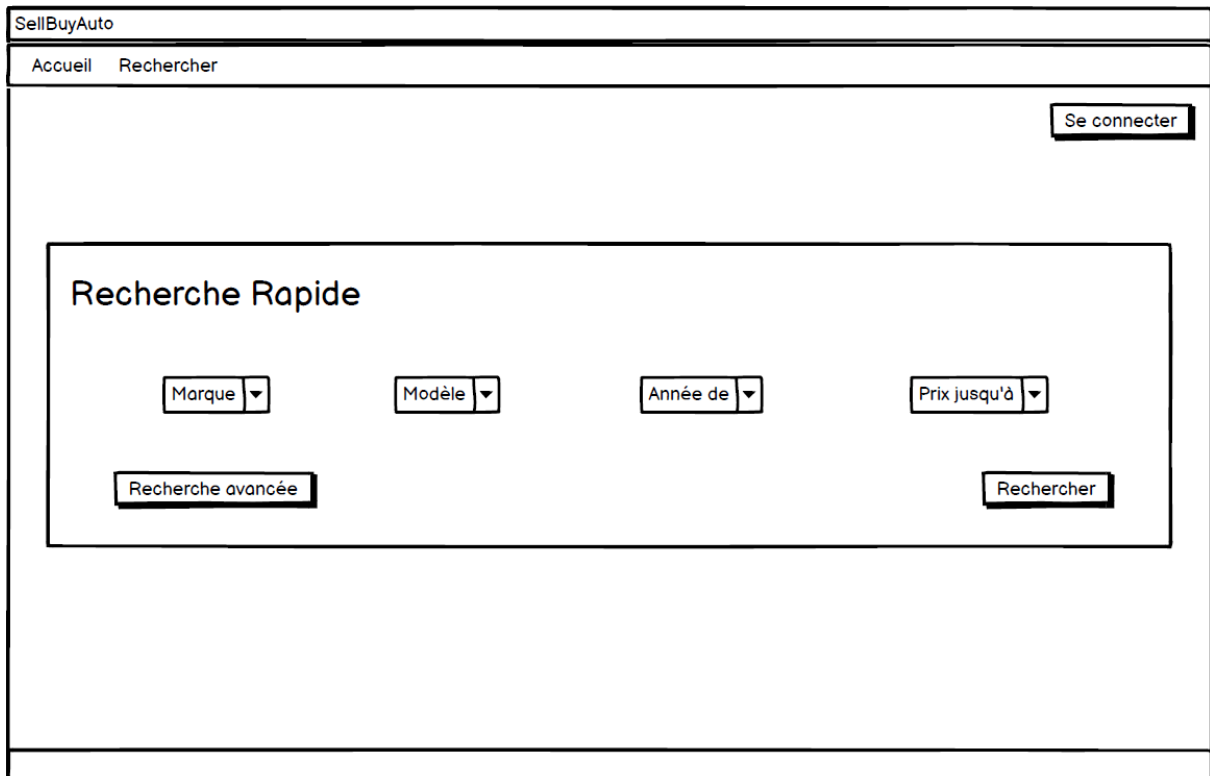


Figure 8 : HomePageWithoutLogin_Wireframe

Cette page correspond à la page d'accueil en n'étant pas connecté. C'est la première page qui s'affiche lors du lancement de l'application.

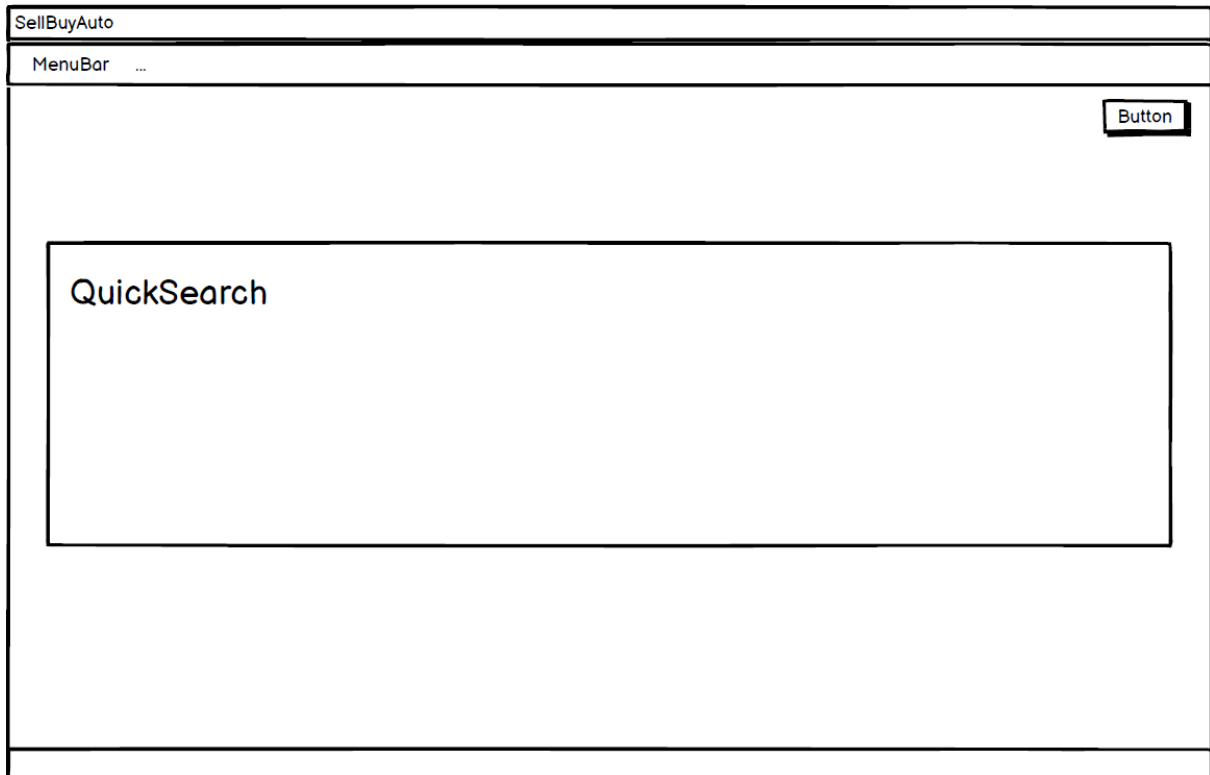


Figure 9 : HomePageWithLogin_Zonning

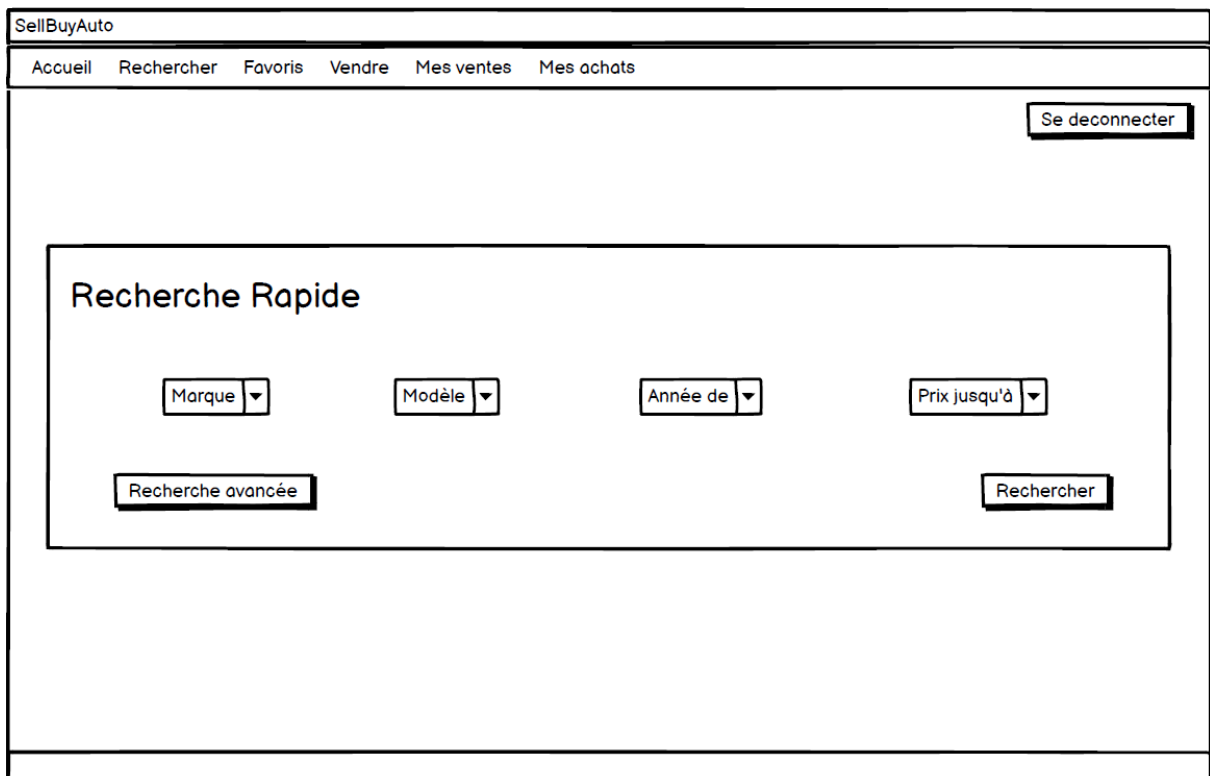


Figure 10 : HomePageWithLogin_Wireframe

Cette page correspond à la page d'accueil une fois que l'utilisateur sera connecté.

The diagram shows a window titled "SellBuyAuto". At the top, there is a "MenuBar" with an ellipsis "...". The main content area contains a large rectangle labeled "Search" and a "Button" located in the top right corner of the window.

Figure 11 : SearchPage_Zonning

The diagram shows a window titled "SellBuyAuto". The top navigation bar includes links: "Accueil", "Rechercher", "Favoris", "Vendre", "Mes ventes", and "Mes achats". A "Se deconnecter" button is in the top right. The main content area is titled "Recherche" and contains several filter sections:

- "Marque" and "Modèle" as dropdown menus.
- "Motorisation" as a dropdown menu.
- "Année" with a range from 1900 to 2024.
- "Kilométrage" with a range from 0 to 400000+.
- "Prix" with a range from 0 to 1000000+.

 A "Rechercher" button is located at the bottom right of the search filters.

Figure 12 : SearchPage_Wireframe

Cette page correspond à la page de recherche avancée. Elle permettra d'effectuer des recherches avec plus de critères que la recherche de la page d'accueil.

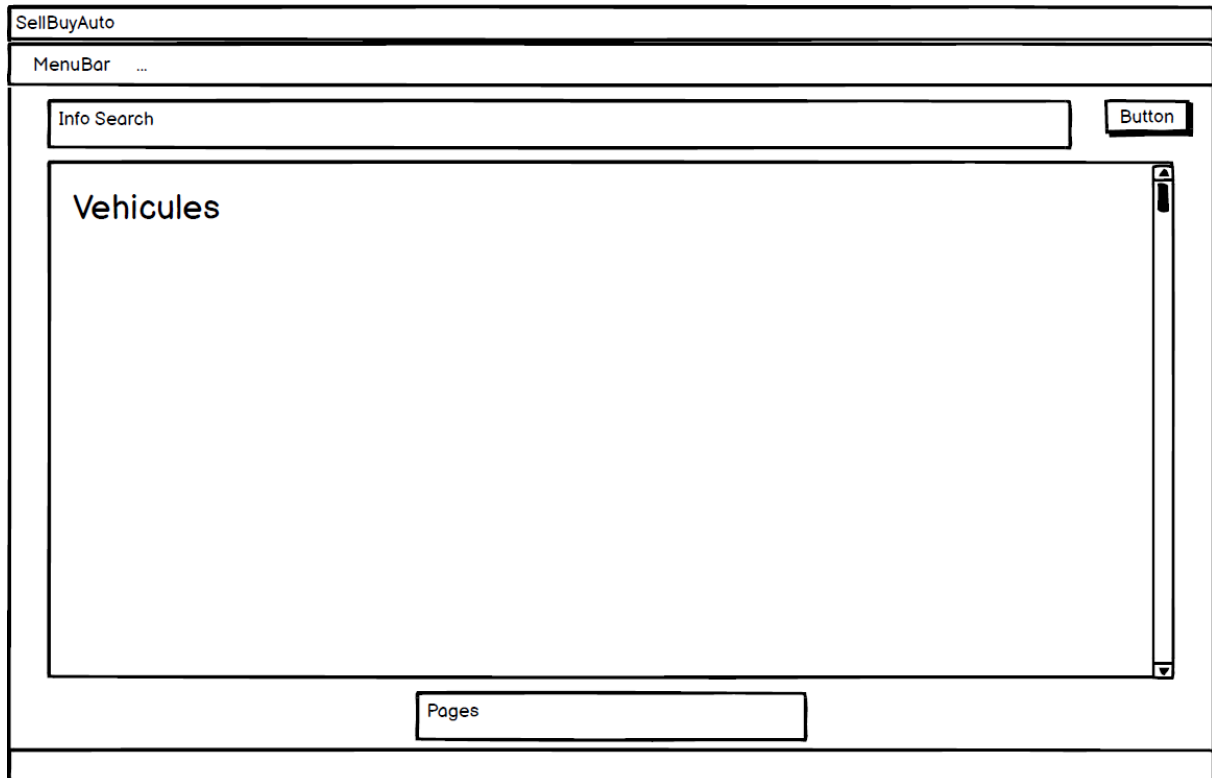


Figure 13 : VehiculesSearchPage_Zonning

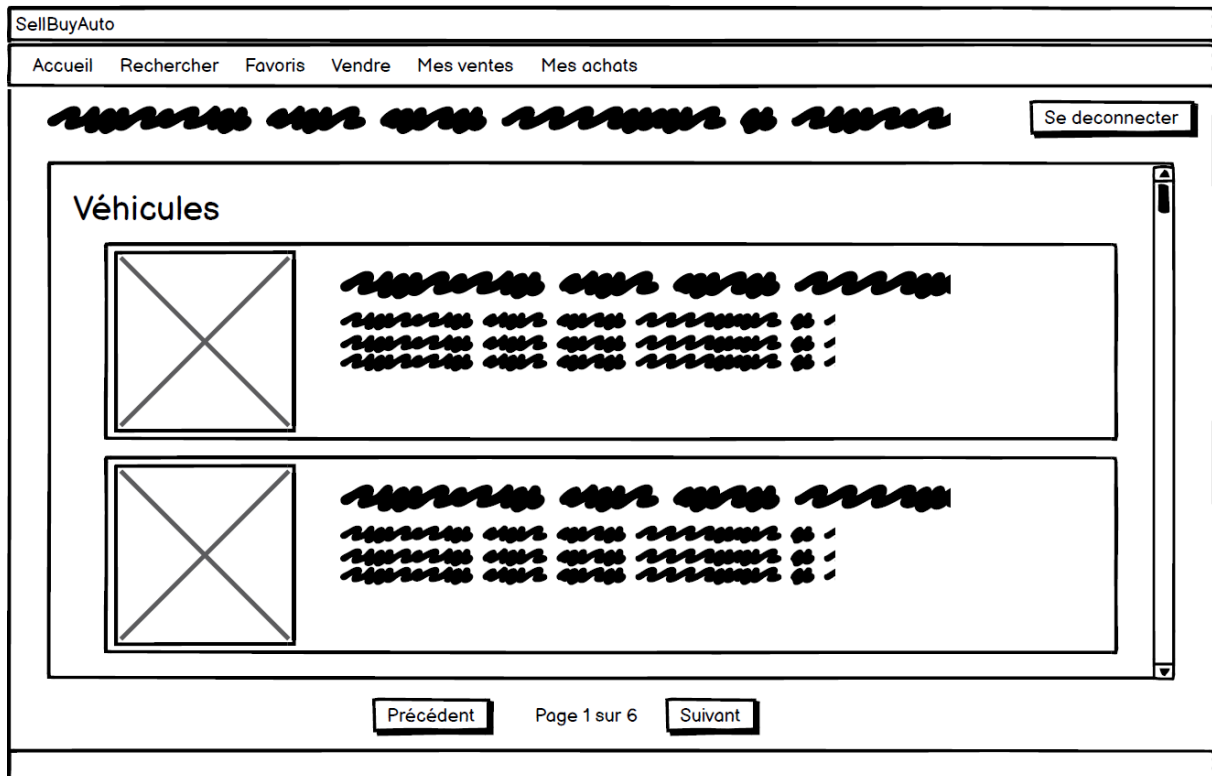


Figure 14 : VehiculesSearchPage_Wireframe

Cette page correspond à la page d'affichage des annonces recherchées.

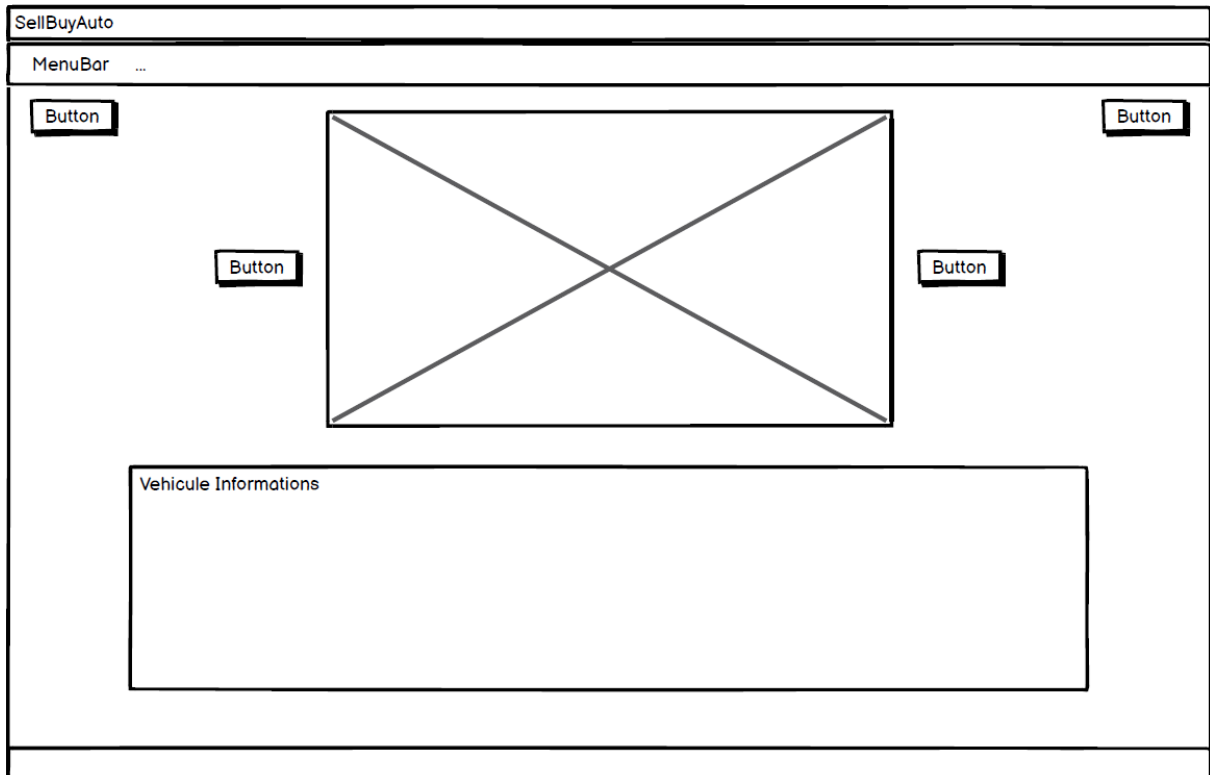


Figure 15 : VehiculeDetailPage_Zonning

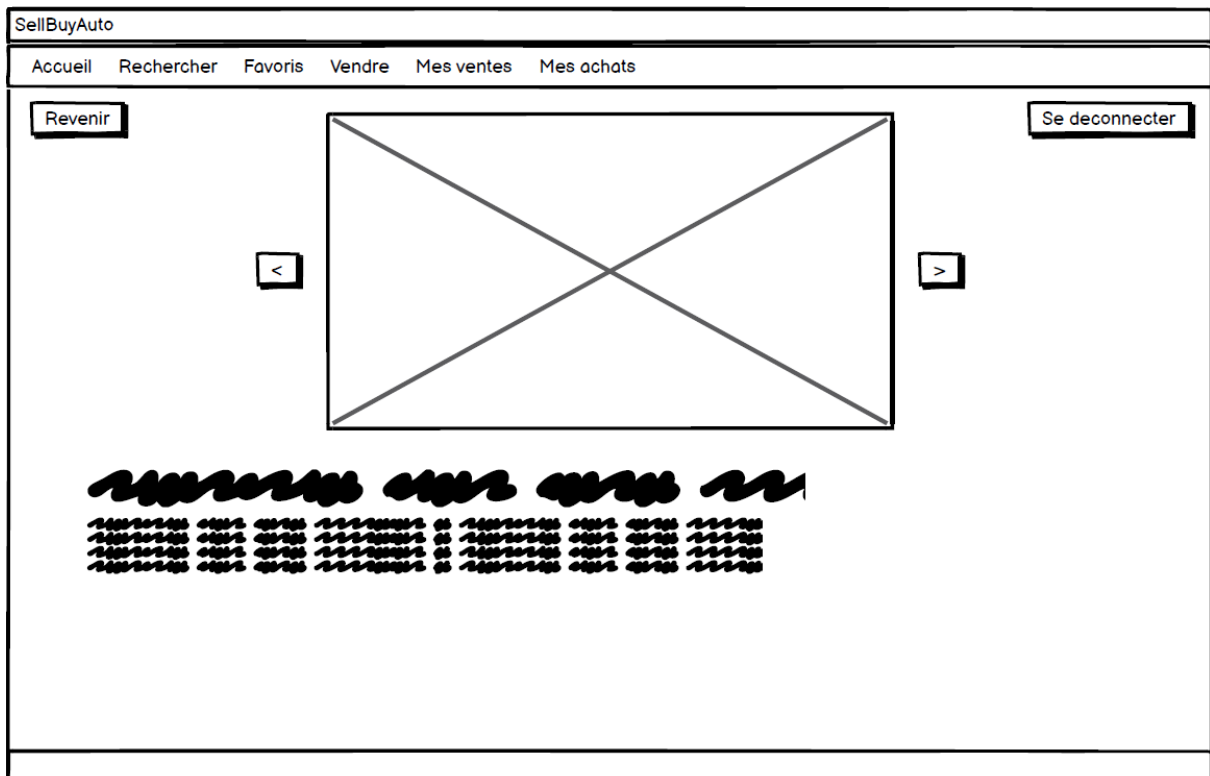


Figure 16 : VehiculeDetailPage_Wireframe

Cette page correspond à la page d'affichage du détail d'une annonce. À condition que l'utilisateur soit connecté, cette page lui permettra de contacter le vendeur.

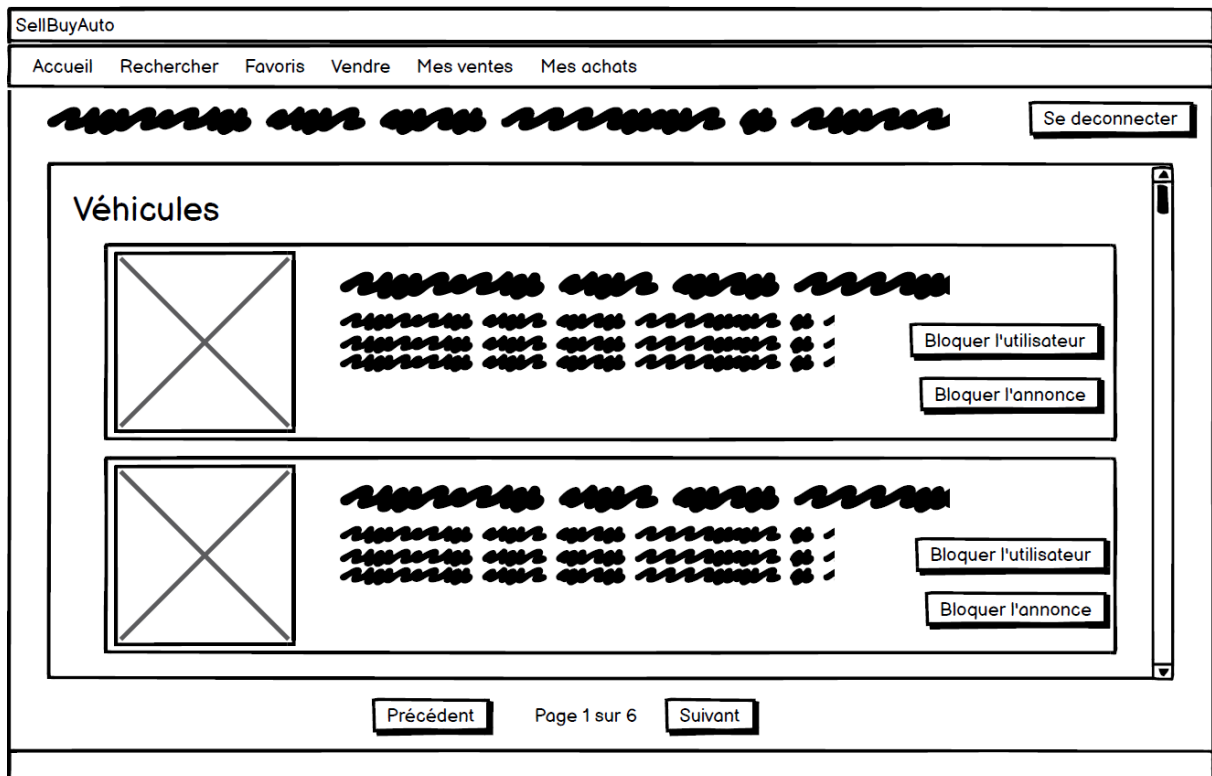


Figure 17 : VehiculesSearchPageInAdmin_Wireframe

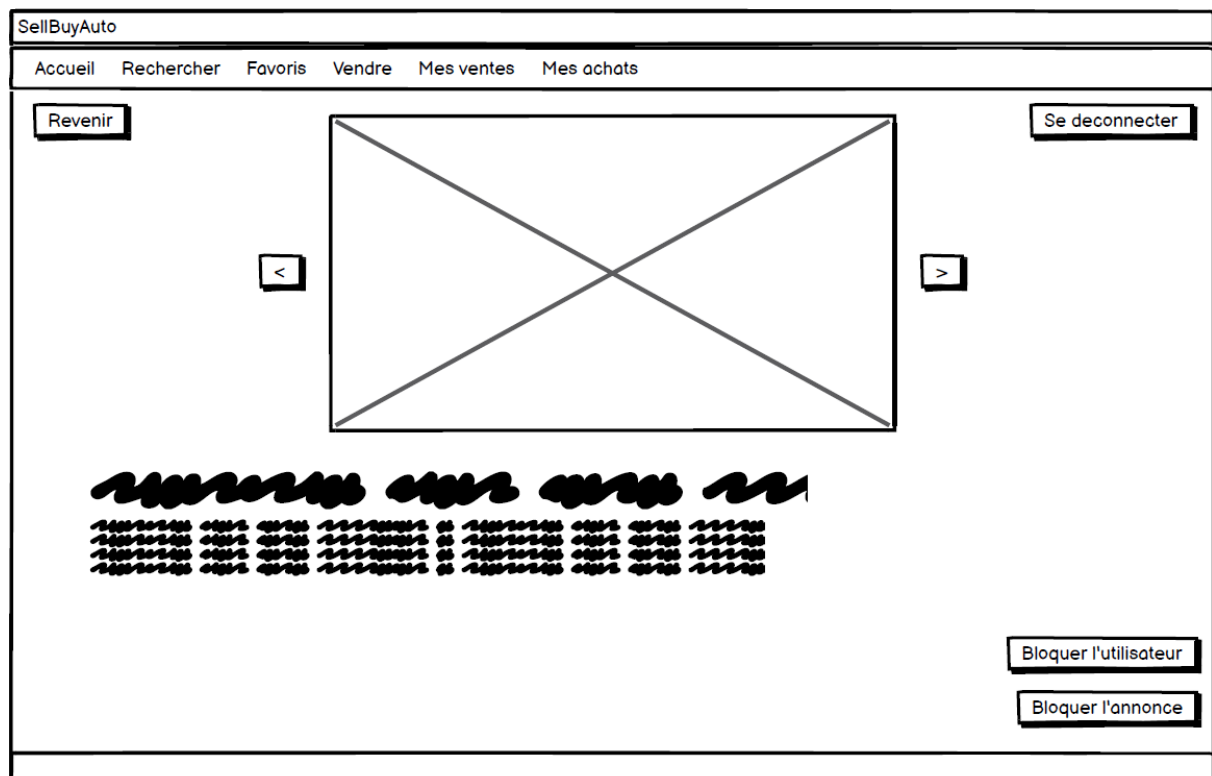


Figure 18 : VehiculeDetailPageInAdmin_Wireframe

Ces pages correspondent respectivement à la page de l'affichage de recherche et à la page du détail d'une annonce lorsque l'utilisateur se connecte en tant qu'administrateur.

The diagram shows a window titled "SellBuyAuto". Below the title bar is a "MenuBar" with an ellipsis "...". The main content area contains a large rectangular box labeled "Form". In the top right corner of the window, there is a button labeled "Button".

Figure 19 : AddModifyPage_Zonning

The diagram shows a window titled "SellBuyAuto". Below the title bar is a menu bar with the following items: "Accueil", "Rechercher", "Favoris", "Vendre", "Mes ventes", and "Mes achats". The main content area contains a form with the following elements:

- Input fields for "Marque :", "Année :", and "Prix :".
- Input fields for "Modèle :", "Kilométrage", and a "ComboBox" for "Motorisation :".
- A large text area for "Description".
- A button labeled "Charger images" next to the "Prix :" field.
- A button labeled "Mettre en vente" at the bottom right of the form.
- A button labeled "Se deconnecter" in the top right corner of the window.

Figure 20 : AddModifyPage_Wireframe

Cette page correspond à la page d'ajout et de modification d'une annonce. Elle permettra à l'utilisateur qui est connecté de créer ou de modifier une annonce.

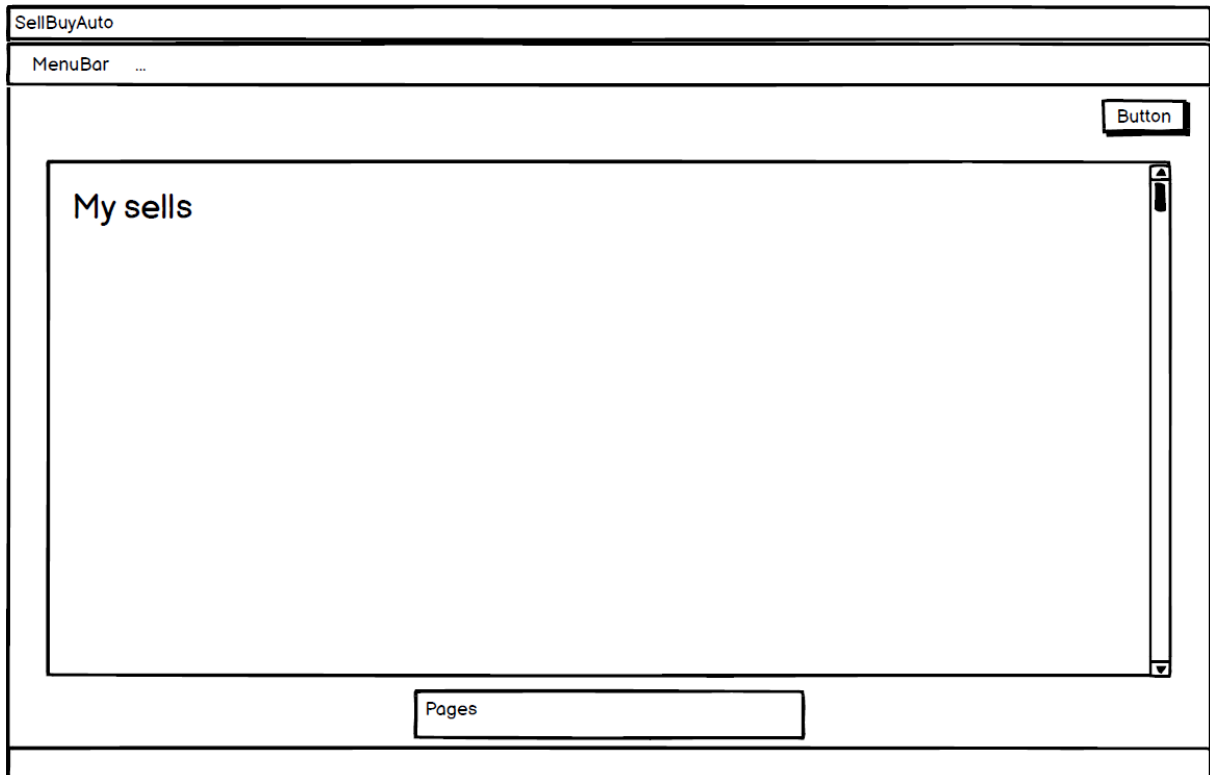


Figure 21 : MySellsPage_Zonning

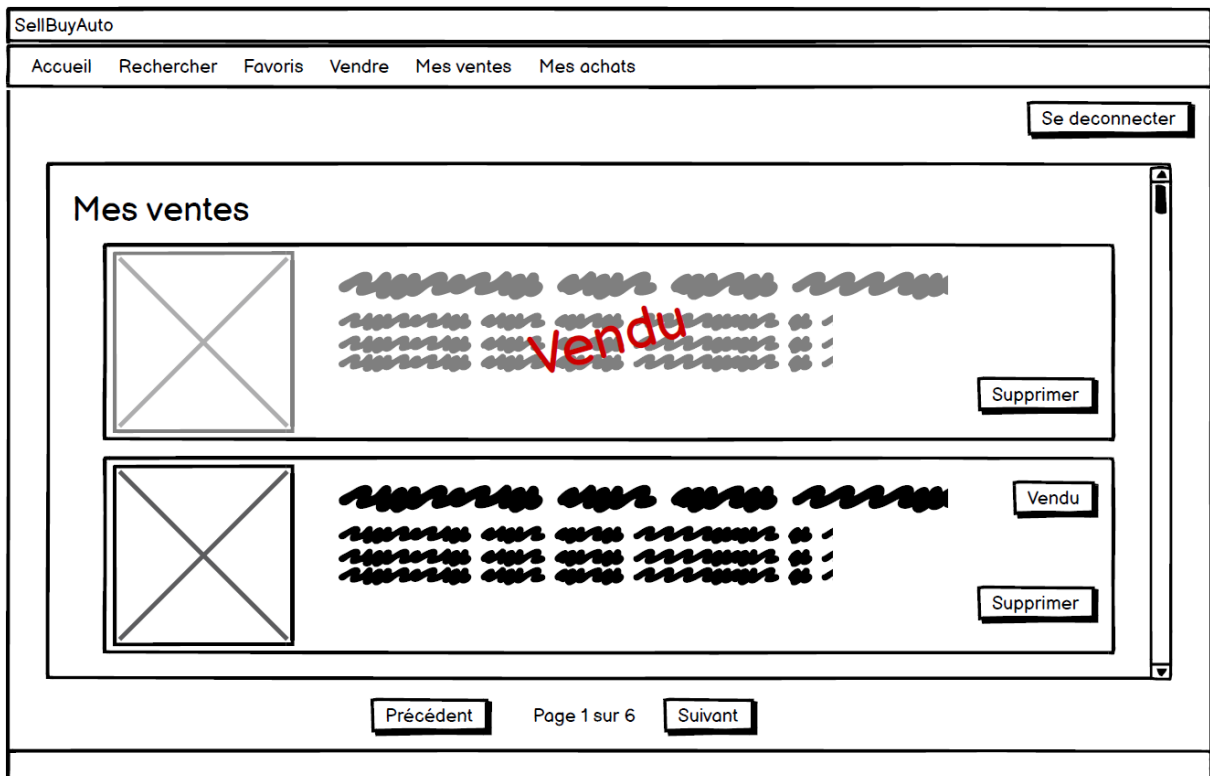


Figure 22 : MySellsPage_Wireframe

Cette page correspond à la page d'affichage des annonces mises en vente par l'utilisateur qui est connecté.

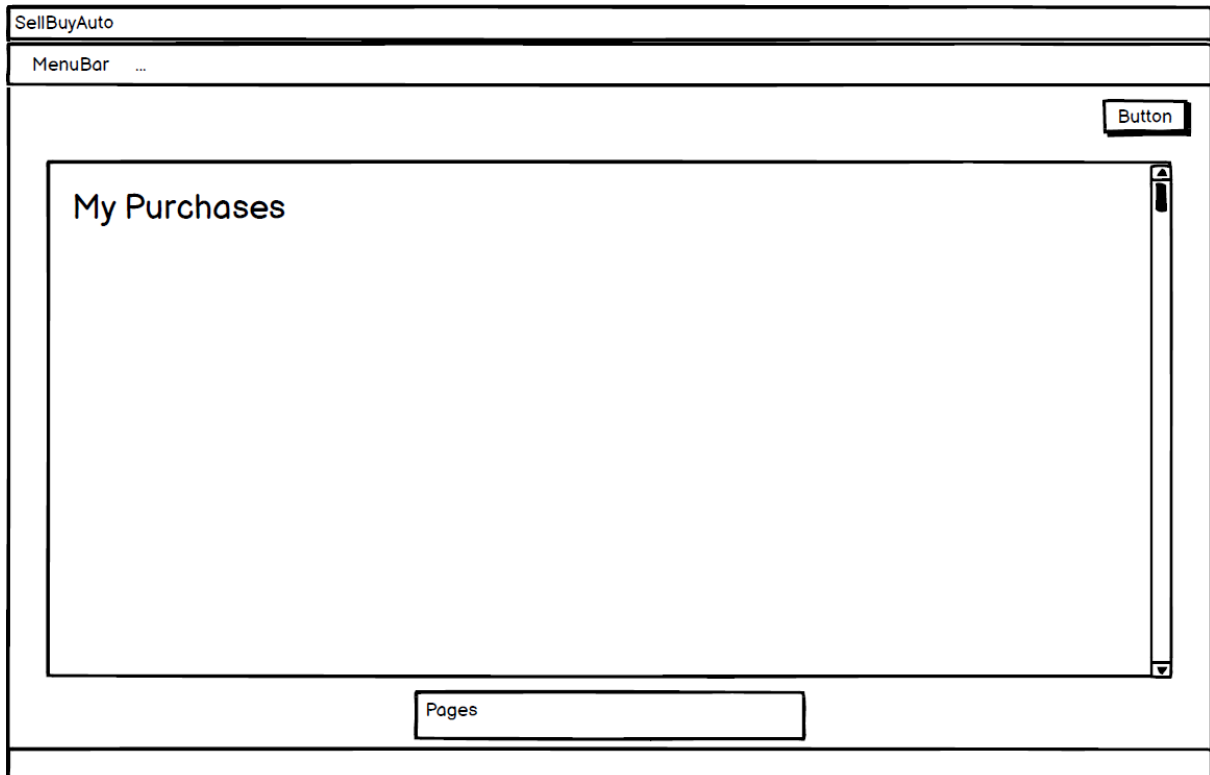


Figure 23 : MyPurchasesPage_Zonning

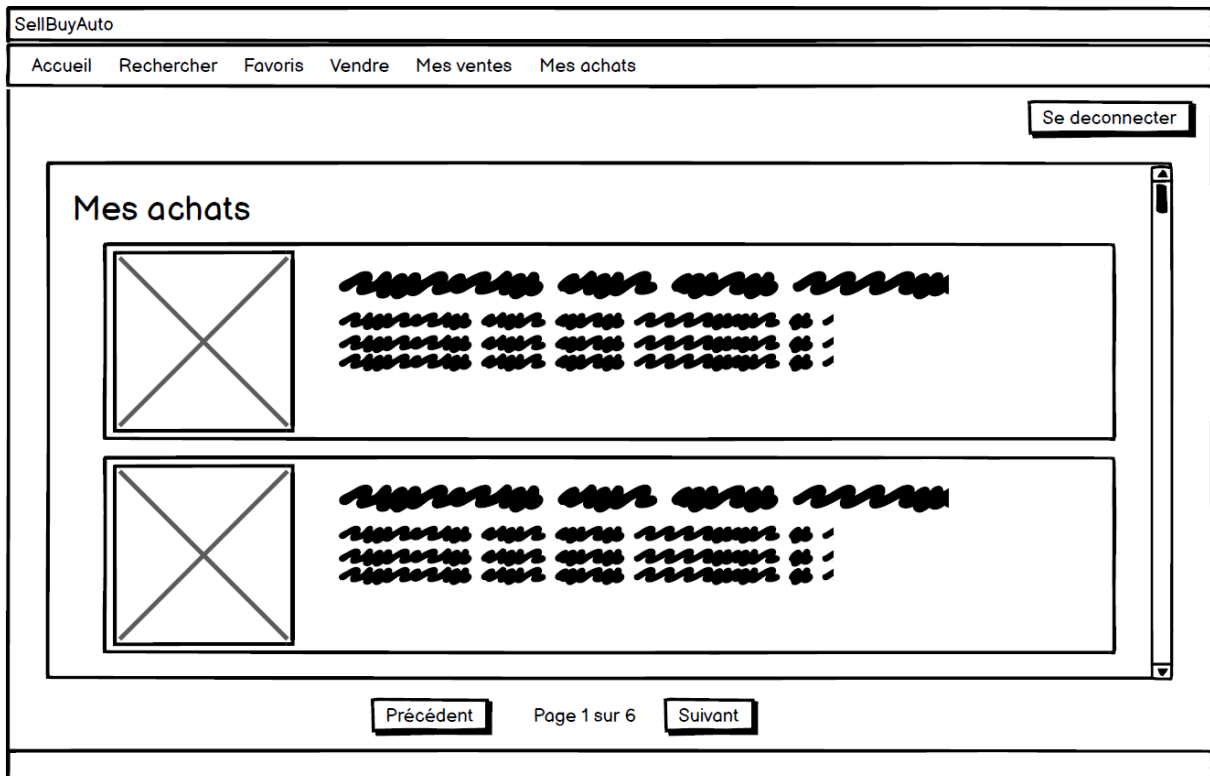


Figure 24 : MyPurchasesPage_Wireframe

Cette page correspond à la page d'affichage des annonces achetées par l'utilisateur qui est connecté.

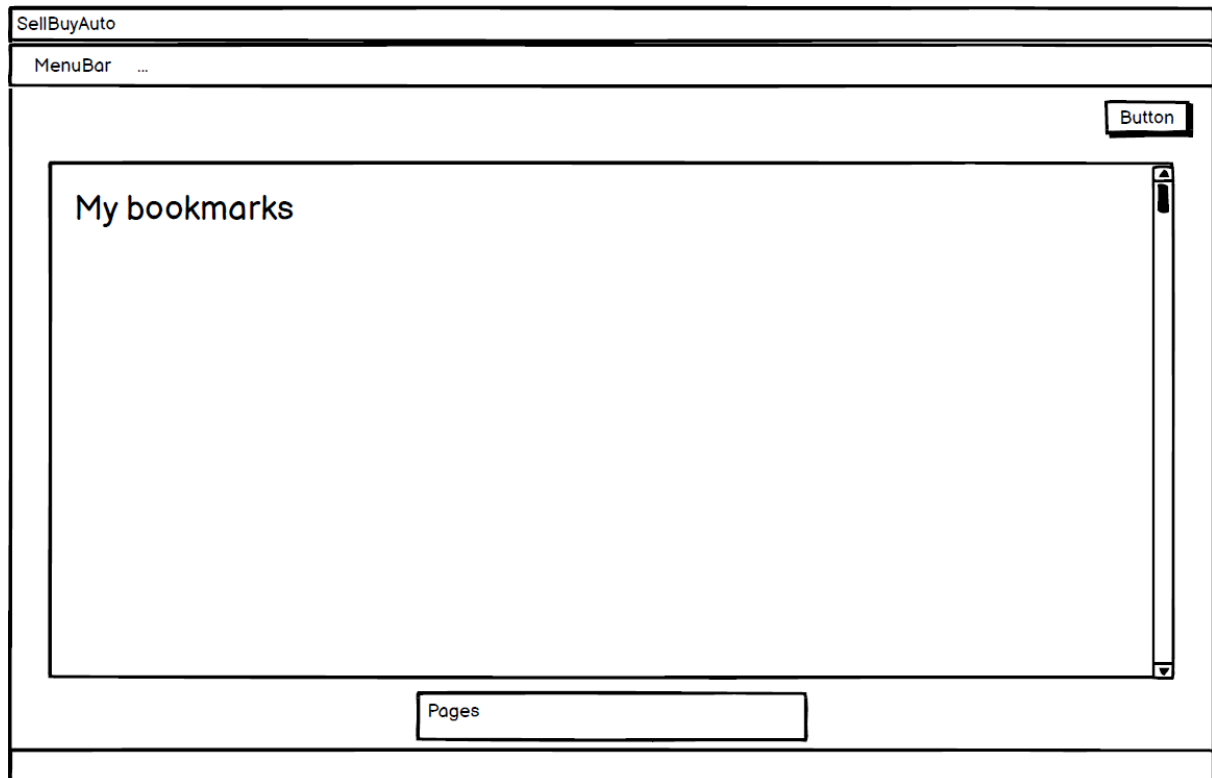


Figure 25 : MyBookmarksPage_Zonning

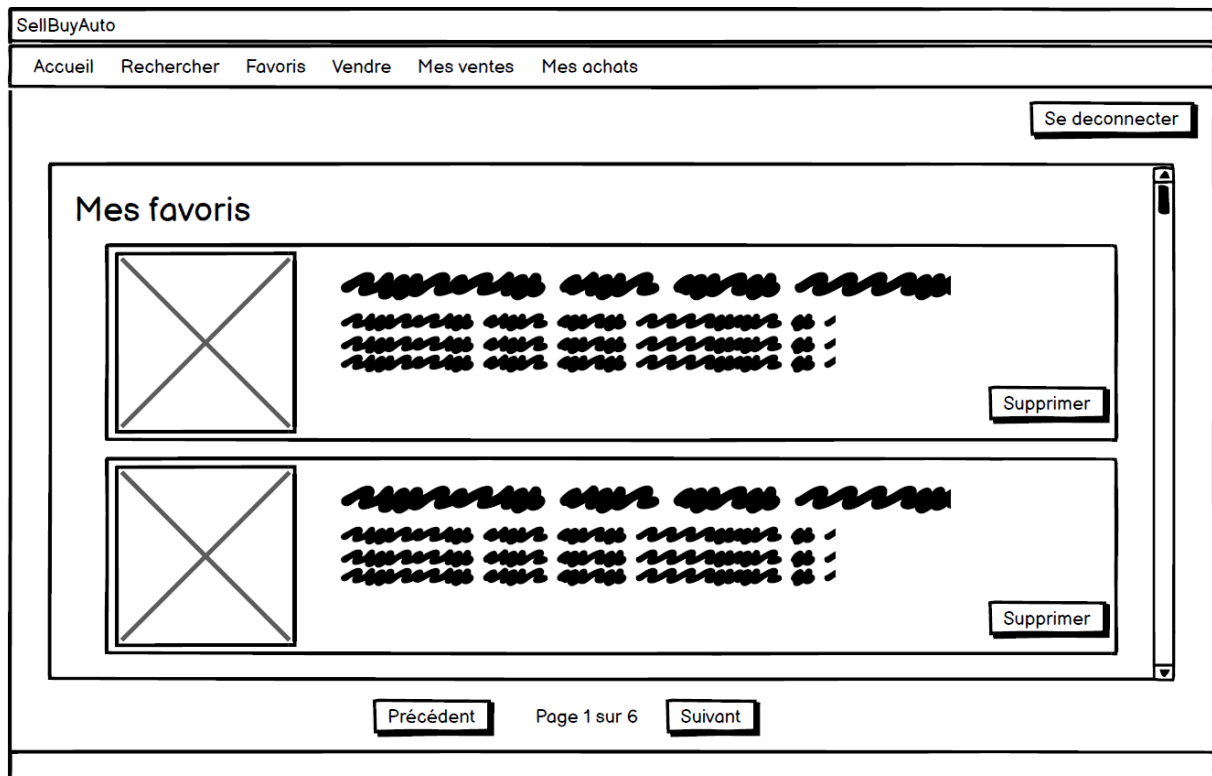


Figure 26 : MyBookmarksPage_Wireframe

Cette page correspond à la page d'affichage des annonces mises en favoris par l'utilisateur qui est connecté.

2.1.3 Use Cases et scénarios

Cas d'Utilisation	Action	Conditions	Réaction
S'enregistrer	Remplir le formulaire d'inscription avec les informations requises	Le nom d'utilisateur n'existe pas déjà dans la base de données	Un nouveau compte utilisateur est créé dans la base de données et l'utilisateur est redirigé vers la page d'accueil avec un message de succès
		Le nom d'utilisateur existe déjà dans la base de données	Un message d'erreur indiquant que le nom d'utilisateur est déjà pris s'affiche
Se connecter	Saisir le nom d'utilisateur et le mot de passe	Les informations de connexion sont correctes	L'utilisateur est connecté à son compte et redirigé vers la page d'accueil
		Les informations de connexion sont incorrectes	Un message d'erreur indiquant que le login n'est pas correct s'affiche
Ajouter une annonce	Remplir les informations de l'annonce (marque, modèle, année, prix, etc.)	Toutes les informations obligatoires sont renseignées	L'annonce est enregistrée dans la base de données et un message de succès s'affiche
		Certaines informations obligatoires ne sont pas renseignées	Un message d'erreur indiquant les champs manquants s'affiche
Modifier une annonce	Sélectionner l'annonce à modifier et modifier les informations nécessaires	Toutes les informations obligatoires sont renseignées	Les informations de l'annonce sont mises à jour dans la base de données et un message de succès s'affiche
		Certaines informations obligatoires ne sont pas renseignées	Un message d'erreur indiquant les champs manquants s'affiche
Supprimer une annonce	Sélectionner l'annonce à supprimer	Aucun utilisateur n'a mis l'annonce en favoris	L'annonce est mise inactive dans la base de données et un message de confirmation s'affiche
		Des utilisateurs ont mis l'annonce en favoris	Une confirmation est demandée à l'utilisateur avant de supprimer l'annonce
Mettre une annonce en favoris	Cliquer sur l'étoile à côté de l'annonce	L'utilisateur est connecté à son compte	L'annonce est ajoutée aux favoris de l'utilisateur

		L'utilisateur n'est pas connecté à son compte	Aucune étoile n'est affichée
Consulter les annonces favorites	Cliquer sur le bouton « Favoris » dans le menu	L'utilisateur a des annonces en favoris	La liste des annonces favorites de l'utilisateur s'affiche
		L'utilisateur n'a pas d'annonces en favoris	Un message d'erreur indiquant qu'il n'y a pas d'annonces favorites s'affiche
Consulter ses ventes	Cliquer sur le bouton « Mes ventes » dans le menu	L'utilisateur a des annonces mises en vente	La liste des annonces mises en vente par l'utilisateur s'affiche
		L'utilisateur n'a pas d'annonces mises en vente	Un message d'erreur indiquant qu'il n'y a pas d'annonces mises en vente s'affiche
Consulter ses achats	Cliquer sur le bouton « Mes achats » dans le menu	L'utilisateur a des annonces achetées	La liste des annonces achetées par l'utilisateur s'affiche
		L'utilisateur n'a pas d'annonces achetées	Un message d'erreur indiquant qu'il n'y a pas d'annonces achetées s'affiche
Bloquer un utilisateur	Sélectionner l'utilisateur à bloquer	L'utilisateur est un administrateur	L'utilisateur est mis comme bloqué dans la base de données et un message de confirmation s'affiche
Bloquer une annonce	Sélectionner l'annonce à bloquer	L'utilisateur est un administrateur	L'annonce est mise comme bloquée dans la base de données et un message de confirmation s'affiche

Tableau 1 : Cas d'utilisation et scénarios

2.2 Stratégie de test

Dans ce projet, la stratégie de test consistera à réaliser des tests au fur et à mesure de l'avancement du projet. Toutes les fonctionnalités ainsi que les cas d'utilisation seront testés de manière individuelle afin de vérifier que le résultat obtenu correspond aux attentes et pour identifier d'éventuelles erreurs de fonctionnement. À la fin de chaque sprint, des tests de non-régression seront effectués afin de garantir que les nouvelles fonctionnalités n'impactent pas le bon fonctionnement des anciennes. De plus, je souhaiterais réaliser une phase de test finale impliquant un utilisateur externe non familier avec le projet afin d'obtenir un retour objectif sur l'expérience utilisateur et identifier d'éventuelles erreurs restantes.

2.3 Planification

2.3.1 Sprint 1

Dates : du 30 avril au 6 mai 2024

Sprint Goal : À la fin du sprint 1, j'aimerais que mon environnement de développement soit mis en place. De plus, je souhaiterais avoir établi les différentes maquettes de l'application et les diagrammes qui me permettront de débiter le développement. Enfin, j'aimerais également que ma base de données soit opérationnelle.

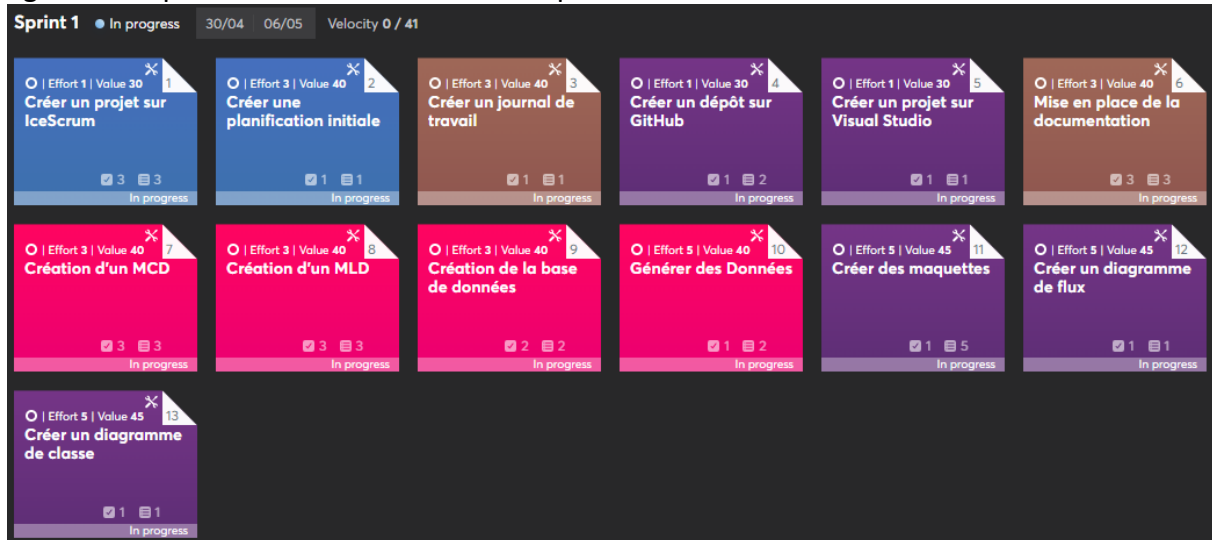


Figure 27 : Stories du sprint 1

2.3.2 Sprint 2

Dates : du 7 mai au 16 mai 2024

Sprint Goal : À la fin du sprint 2, j'aimerais avoir fini le développement de la partie publique de l'application qui consiste à voir les véhicules en vente, et faire des recherches avec des filtres. J'aimerais également commencer le développement de la partie vente/achat.

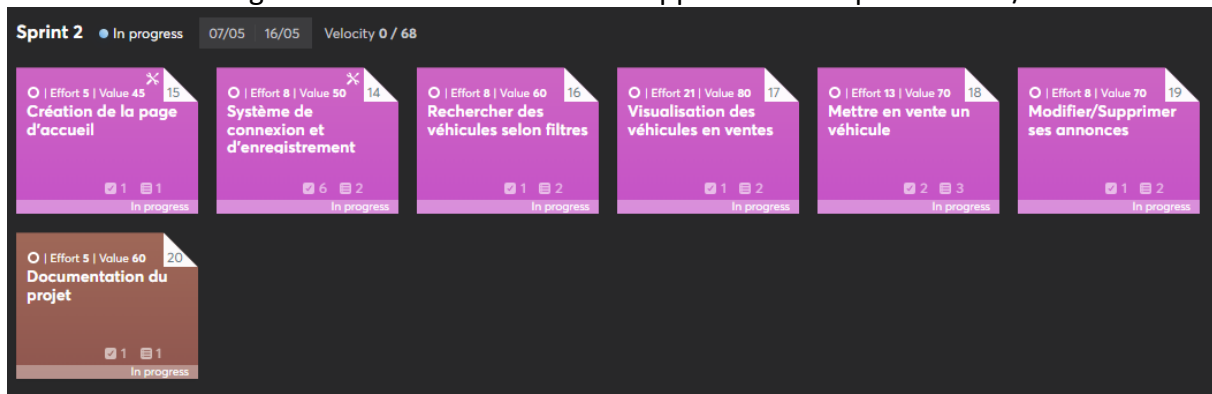


Figure 28 : Stories du sprint 2

2.3.3 Sprint 3

Dates : du 17 mai au 23 mai 2024

Sprint Goal : À la fin du sprint 3, j'aimerais avoir fini le développement de la partie vente/achat.

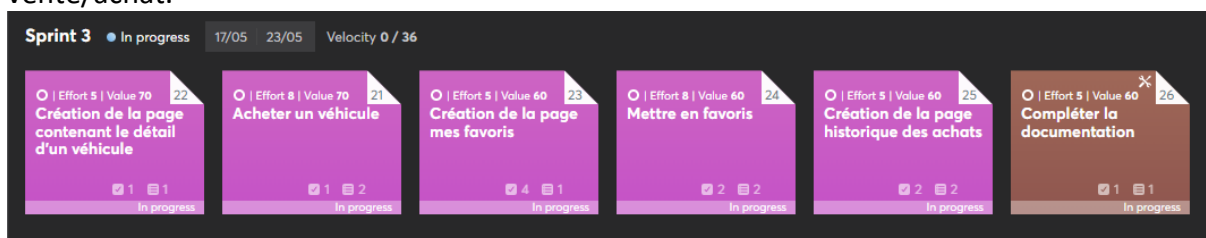


Figure 29 : Stories du sprint 3

2.3.4 Sprint 4

Dates : du 24 mai au 29 mai 2024

Sprint Goal : À la fin du sprint 4, j'aimerais avoir fini le développement de la partie administration, et donc de l'application, ainsi que la documentation du projet.

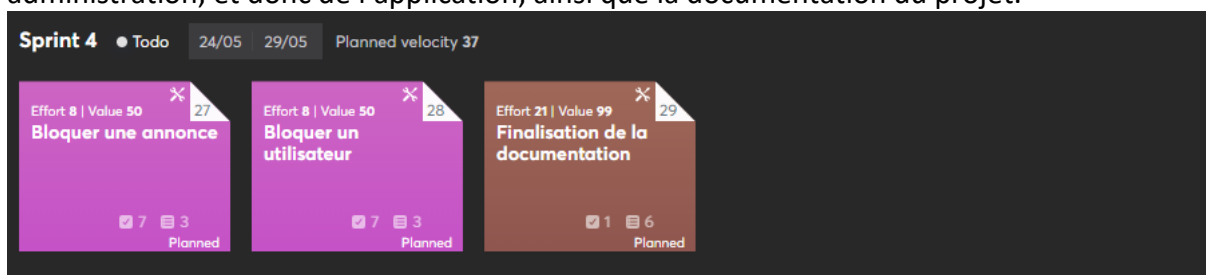


Figure 30 : Stories du sprint 4

3 Conception

3.1 Logiciels

Pour ce projet, j'ai opté pour Visual Studio Community 2022 comme principal environnement de développement pour le code. J'ai créé un projet de type « Windows Form », me permettant ainsi de concevoir une application avec une interface graphique plutôt qu'une application de type console. Le Framework utilisé pour le projet est .NET 8.0.

J'ai choisi ce Framework car c'est la dernière version stable et elle sera prise en charge sur le long terme.

Pour la création du diagramme de classes ainsi que pour l'élaboration du Modèle Conceptuel de Données (MCD), j'ai utilisé le logiciel Draw.io. Pour le Modèle Logique de Données (MLD), j'ai utilisé le logiciel MySQL Workbench, qui m'a permis de générer un fichier SQL pour la création de la base de données à partir du MLD.

Pour la gestion de la base de données, j'ai fait le choix d'utiliser les serveurs SwissCenter pour stocker les données ainsi que les images, et j'accède à ces données de manière graphique à l'aide de HeidiSQL. Ce logiciel me permet également de créer des sauvegardes de ma base de données. Pour accéder aux images, j'utilise une connexion FTP via le logiciel Total Commander.

En ce qui concerne la conception des maquettes de l'application, j'utilise Balsamiq.

3.2 Choix techniques

Pour simplifier le processus d'ajout et de modification d'annonces, les utilisateurs doivent saisir la marque et le modèle des véhicules, étant donné que je ne dispose pas d'une base de données contenant l'intégralité des marques et modèles des voitures. Cependant, lorsqu'ils effectuent une recherche, une liste déroulante leur permet de sélectionner la marque et le modèle parmi les voitures disponibles à la vente.

La gestion des ventes est également simplifiée. Lorsqu'un vendeur conclut une vente, il doit simplement indiquer l'adresse mail de l'acheteur. L'annonce correspondante est alors automatiquement déplacée vers la section « Mes achats » dans le compte de l'acheteur.

En cas de blocage d'une annonce par un administrateur, celle-ci devient uniquement visible par le vendeur concerné avec un message indiquant que l'annonce est bloquée. De plus, elle est automatiquement retirée des favoris des autres utilisateurs. Cette mesure vise à garantir la sécurité des utilisateurs en éliminant toute annonce potentiellement dangereuse de leur liste de favoris.

Si un vendeur décide de supprimer son annonce, celle-ci disparaît de son compte. Cependant, si la voiture a déjà été vendue, elle reste accessible à l'acheteur dans la section « Mes achats ». Si un utilisateur a ajouté la voiture à ses favoris, une confirmation est demandée au vendeur avant la suppression de l'annonce. Si le vendeur confirme la suppression, l'annonce est retirée des favoris de l'utilisateur.

3.3 MLD

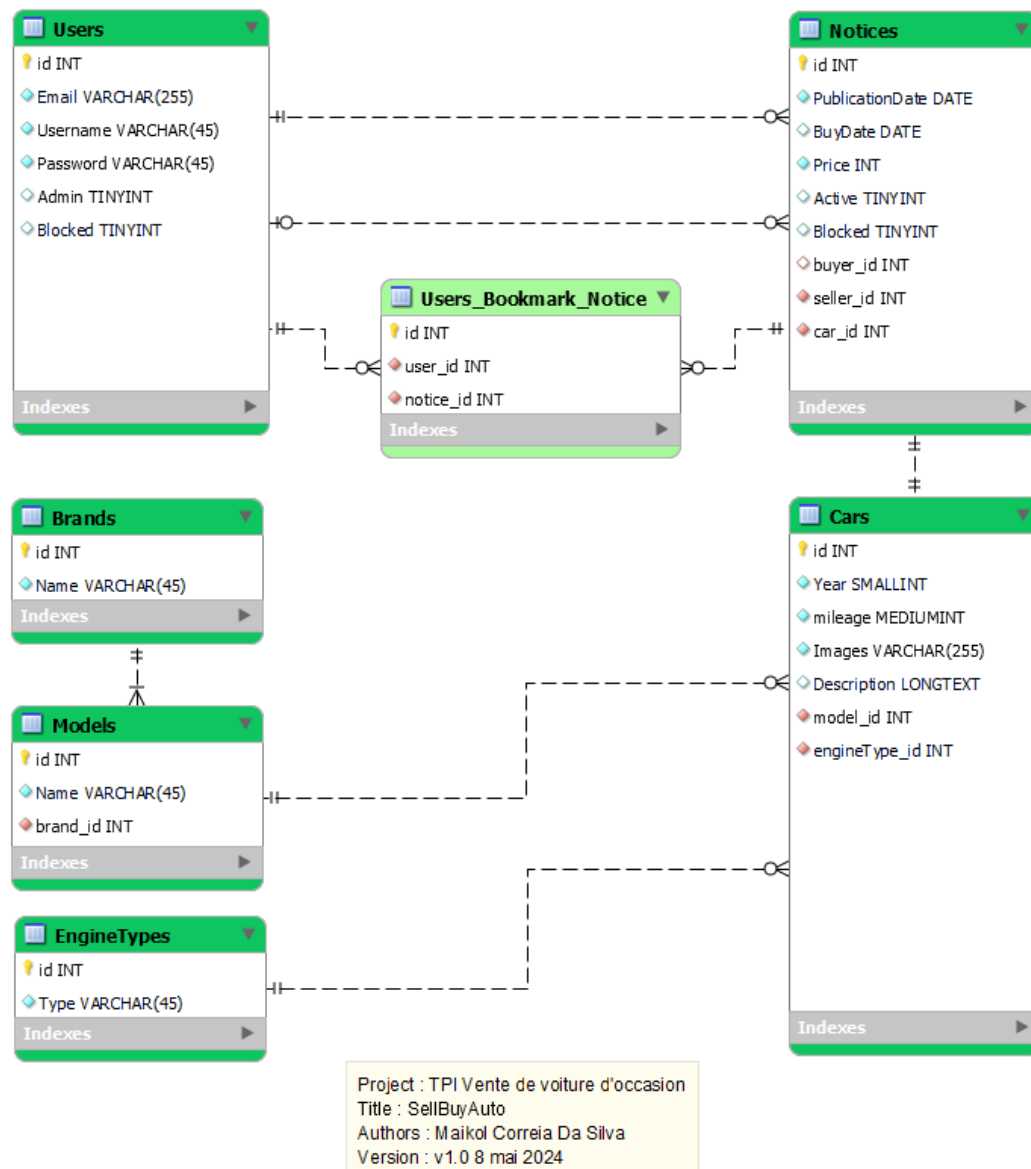


Figure 31 : MLD

Ce Modèle Logique de Données (MLD) a été élaboré à partir du Modèle Conceptuel de Données (MCD) présenté au point **2.1.1 MCD**. Il comprend sept tables au total, dont une table intermédiaire, entre les tables « Users » et « Notices », destinée à enregistrer les annonces mises en favoris par les utilisateurs.

Dans la table « Notices », plusieurs clés étrangères sont présentes. La première, « buyer_id », fait référence à la table « Users » et permet d'identifier l'acheteur du véhicule si celui-ci a été vendu. La deuxième clé, « seller_id », renvoie également à la table « Users » et identifie le vendeur du véhicule concerné. Enfin, la clé « car_id » se rapporte à la table « Cars » et permet d'associer le véhicule mis en vente à l'annonce correspondante.

En complément, trois tables supplémentaires sont présentes : « Brands », « Models », et « EngineTypes ». Ces tables servent à réaliser des listes déroulantes incluant des critères liés aux voitures, tels que la marque ou le modèle, facilitant ainsi la saisie des données lors de la recherche des annonces.

3.4 Diagramme de classes

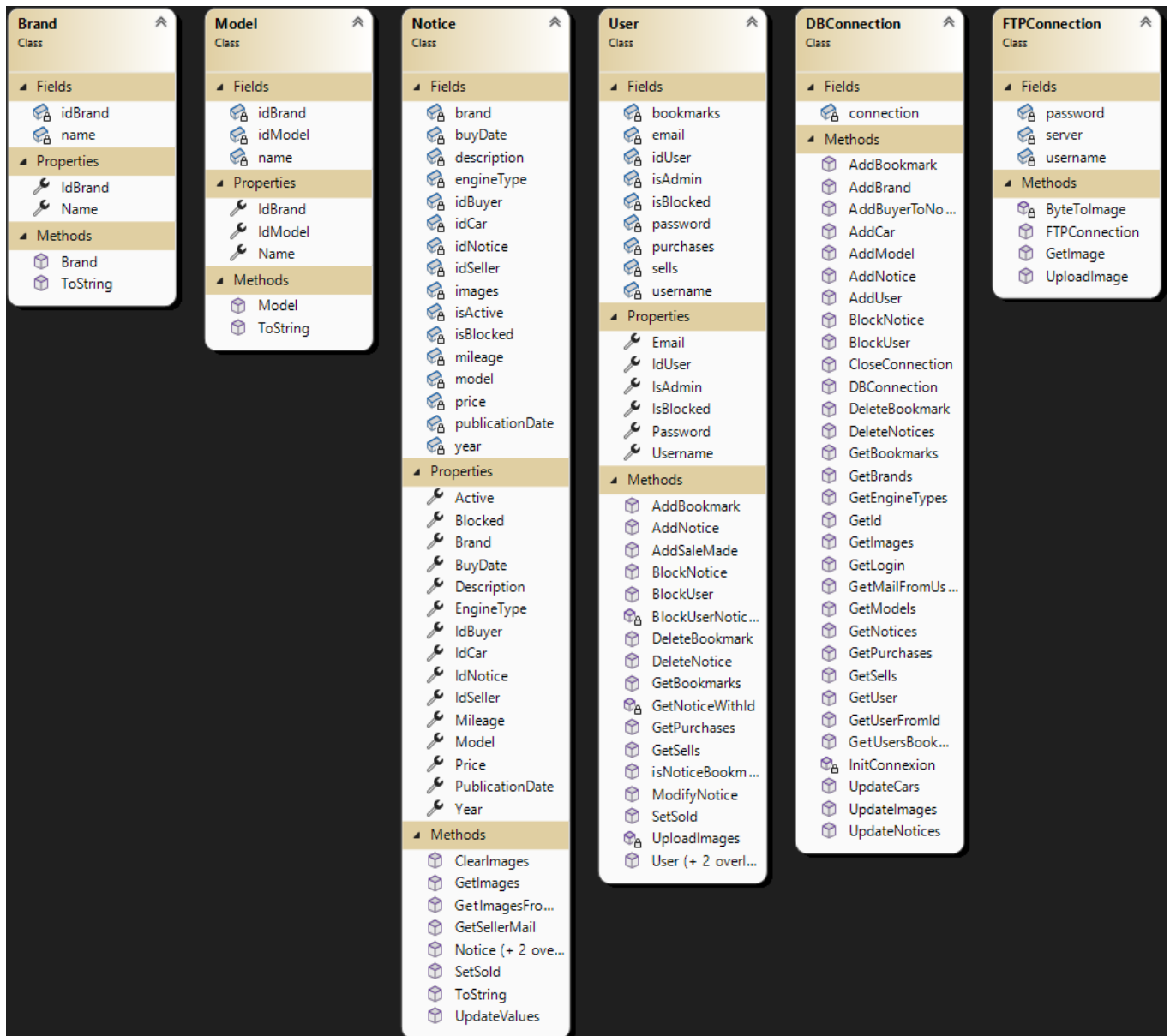


Figure 32 : Diagramme de classes

Ce projet comprend trois classes principales. La première, « Notice », englobe les détails liés aux annonces et aux véhicules. Son rôle principal consiste à créer, modifier et afficher les annonces au sein de l'application. Ensuite, la classe « User » est dédiée à la gestion des informations relatives aux utilisateurs. Elle permet de gérer les véhicules achetés, vendus ou mis en favoris par les utilisateurs. De plus, elle offre aux administrateurs la possibilité de bloquer des utilisateurs ou des annonces. Enfin, la classe « DBConnection » assure la liaison avec la base de données, facilitant ainsi la récupération et l'envoi de données.

3.5 Diagramme de flux

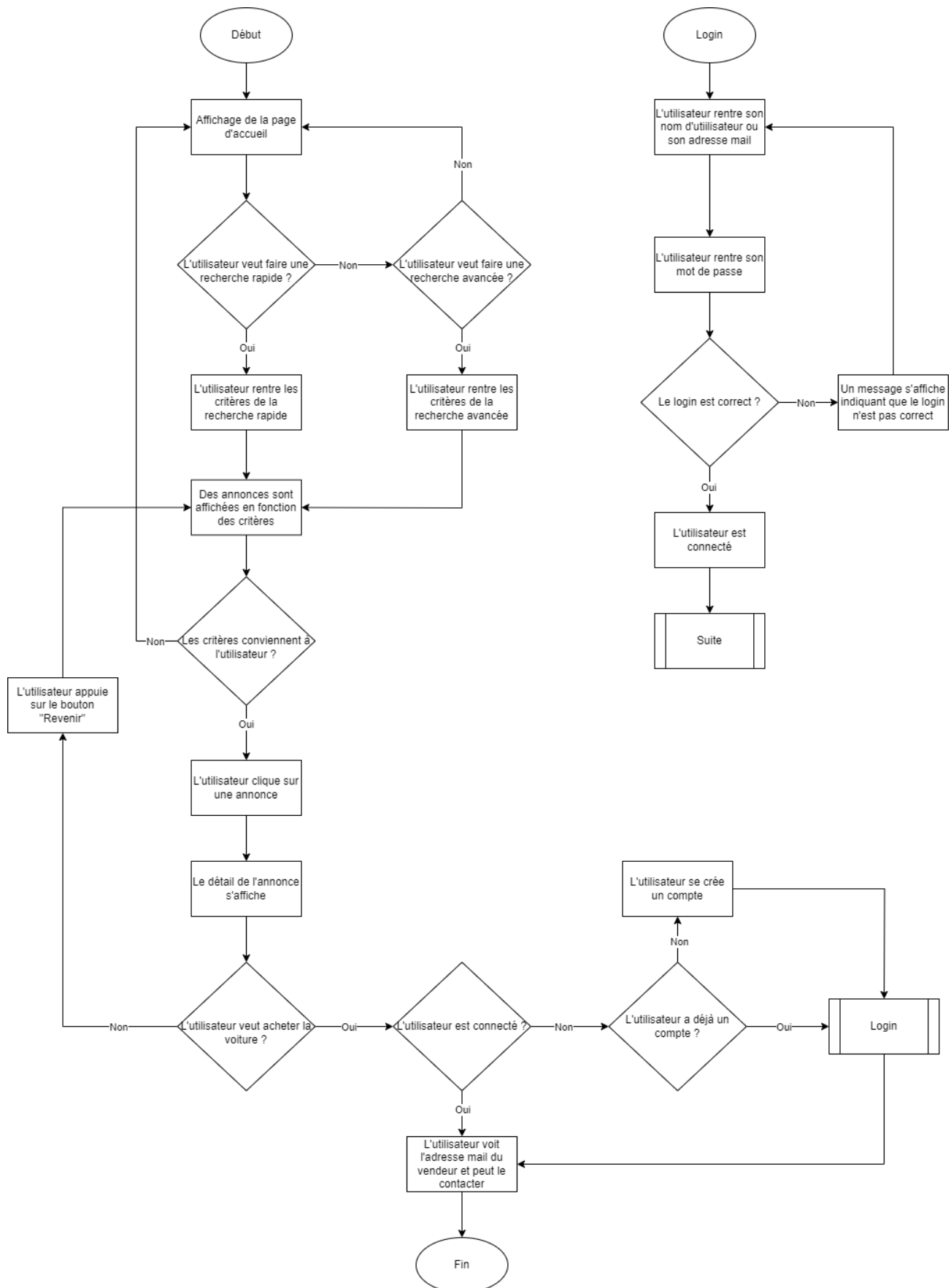


Figure 33 : Diagramme de flux

4 Réalisation

4.1 Dossier de réalisation

La réalisation de l'application s'est déroulée selon la méthode agile. De ce fait, la réalisation a été faite en plusieurs parties, chacune ayant des objectifs spécifiques à atteindre. Voici un aperçu de chaque partie et des tâches accomplies.

4.1.1 Sprint 1

Lors de ce premier sprint, j'ai accompli plusieurs étapes importantes pour l'avancement de mon projet.

Création du projet sur IceScrum : Tout d'abord, j'ai établi un projet IceScrum, où j'ai défini les sprints et les stories qui guideront mon travail tout au long du développement de l'application.

Création d'une planification initiale : Ensuite, j'ai fait une planification initiale pour avoir une vision globale de la répartition des tâches et du déroulement du projet. Cette étape me permet de mieux organiser mon travail par la suite.

Création du projet GitHub et Visual Studio : J'ai par la suite mis en place mon environnement de développement en configurant un dépôt sur GitHub pour versionner mon code et ma documentation. J'ai ensuite initialisé un projet dans Visual Studio.

Création d'un journal de travail : Un journal de travail a également été établi afin de consigner quotidiennement mes activités. Cela m'aidera à suivre ma progression et à rester organisé tout au long du projet.

Mise en place de la documentation : J'ai débuté la documentation du projet, incluant des titres, une introduction et une page de garde, pour fournir un cadre clair et complet pour mon travail.

Création de la base de données : La conception des modèles de données a également été réalisée, avec l'élaboration du Modèle Conceptuel de Données (MCD) et du Modèle Logique de Données (MLD). Ces modèles décrivent la structure et les relations entre les entités de ma base de données que j'ai pu créer. J'ai également généré des données à l'aide de l'IA ChatGPT que j'ai ensuite peaufinées manuellement.

Création des diagrammes : J'ai également créé des diagrammes pour préparer le code de mon application. J'ai donc commencé par un diagramme de flux qui représente le déroulement des fonctionnalités connexion et de recherche d'un véhicule. Puis, j'ai réalisé un diagramme de classes afin de modéliser les différentes classes de mon application.

Création des maquettes : Enfin, j'ai réalisé des maquettes pour concevoir l'interface utilisateur de mon application, me permettant ainsi de planifier et de visualiser son apparence.

En résumé, ce premier sprint a constitué une étape importante dans le lancement de mon projet. Les bases solides ont été posées, ce qui me permettra de poursuivre le développement de manière efficace et structurée.

4.1.2 Sprint 2

Au cours du sprint 2, j'ai entamé la phase de codage de mon application.

Création de la page d'accueil : J'ai tout d'abord commencé avec la création de la page d'accueil dans Windows Form, puis j'ai implémenté plusieurs fonctionnalités. La première était la gestion de la connexion et de l'inscription des utilisateurs, ce qui s'est déroulé sans encombre.

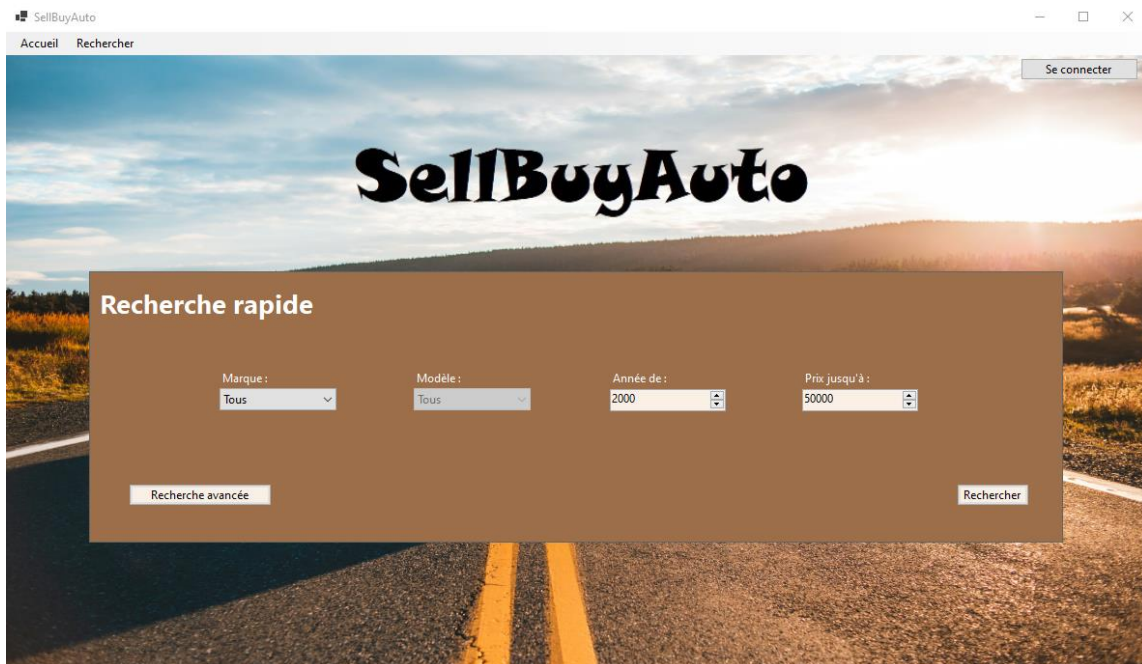


Figure 34 : Page d'accueil

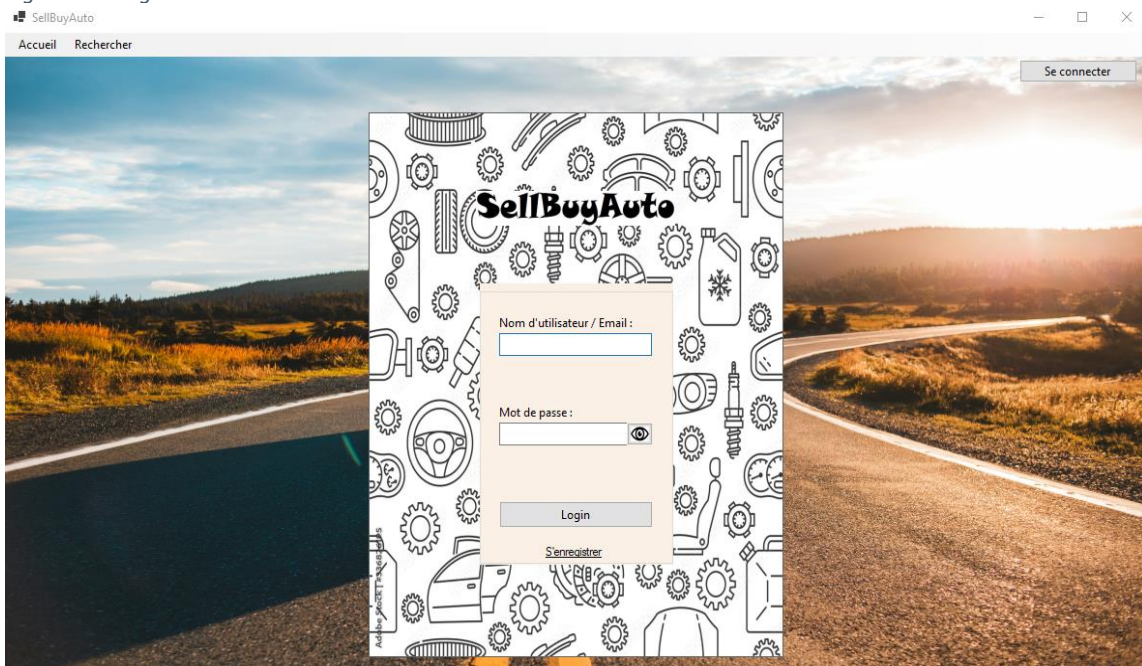


Figure 35 : Page de connexion

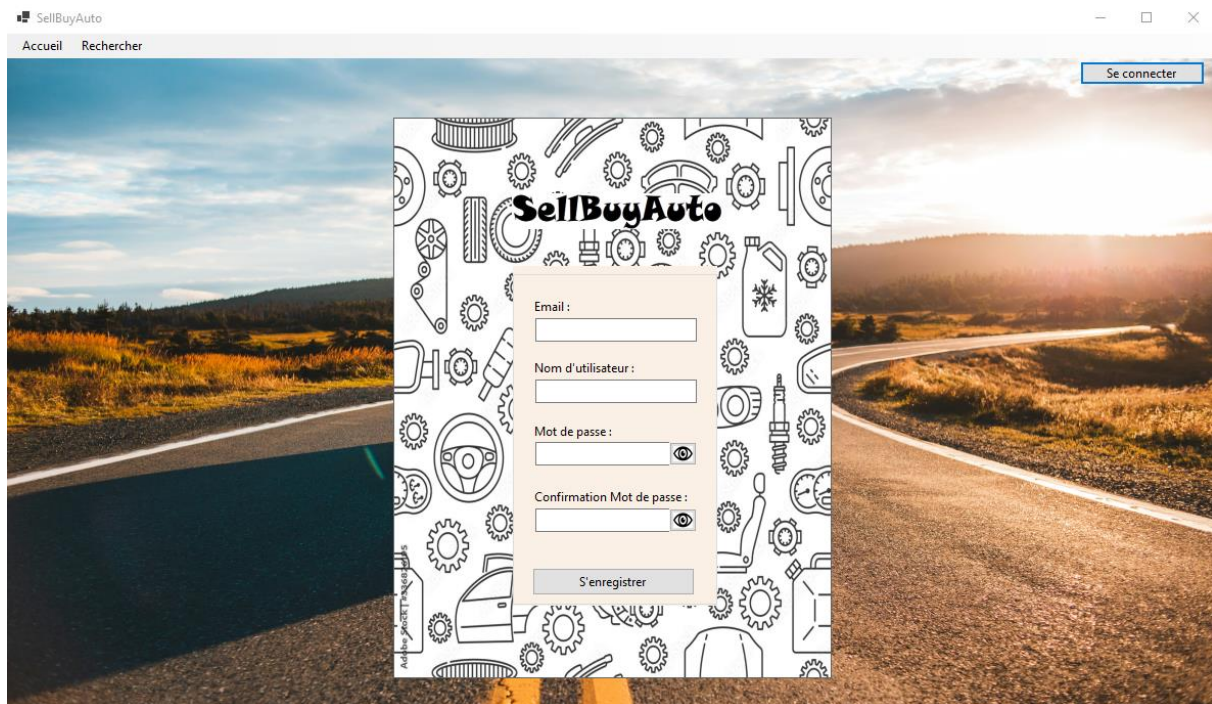


Figure 36 : Page d'enregistrement

Optimisation de l'affichage général : Ensuite, j'ai entamé la fonctionnalité de recherche de véhicules avec des critères spécifiques. Pendant la réalisation de cette dernière, j'ai rapidement réalisé que ma stratégie initiale, consistant à créer plusieurs formulaires distincts, n'était pas optimale. Les transitions entre les pages n'étaient pas fluides, et la présence d'un menu identique sur chaque page allait entraîner une duplication de code inutile. J'ai donc opté pour une approche différente en créant un seul formulaire principal et en utilisant des UserControls pour afficher le contenu dynamiquement. Depuis ce formulaire principal, je fais appel aux UserControls correspondants, et ces derniers envoient des événements au formulaire principal pour gérer la navigation entre les pages.

Utilisation des UserControls : Les UserControls sont des composants réutilisables dans les applications Windows Forms. Ils sont similaires aux contrôles intégrés de base, tels que les boutons, les étiquettes et les zones de texte, mais la seule différence est qu'ils peuvent être créés par les développeurs pour encapsuler des fonctionnalités spécifiques ou des interfaces utilisateur plus complexes.

Dans mon cas, les UserControls constituent une composante essentielle de l'architecture de mon application, offrant une approche modulaire pour la conception de l'interface utilisateur. Chaque UserControl sera conçu pour représenter une fonctionnalité spécifique de l'application et est intégré de manière transparente dans le formulaire principal.

Implémentation de la recherche rapide et avancée : Une fois cette organisation mise en place, j'ai pu finaliser la fonction de recherche avec deux versions : une recherche rapide qui se trouve sur la page d'accueil et une recherche avancée offrant davantage d'options qui se situe sur une page prévue à cet effet.

The screenshot shows the 'Recherche avancée' page with the following search criteria:

- Marque: Tous
- Modèle: Tous
- Motorisation: Tous
- Année: De 1900 à 2024
- Kilométrage: De 0 à 400000
- Prix: De 0 à 1000000

Figure 37 : Page de recherche avancée

Affichage des annonces : J'ai également créé un UserControl qui permet d'afficher les annonces recherchées, en utilisant un contrôle qui s'appelle « FlowLayoutPanel », qui permet un affichage fluide avec un défilement automatique si nécessaire.

The screenshot shows the search results for the criteria defined in Figure 37. The results are as follows:

Image	Titre	Kilométrage	Année	Prix	Description
	Audi A4	11000	2020	40000	L'Audi A4 est une berline familiale du constructeur automobile allemand Audi.
	Mercedes-Benz Classe E	10000	2021	60000	La Mercedes-Benz Classe E est une gamme de voitures exécutives produites par le constructeur automobile allemand Mercedes-Benz.

Figure 38 : Page d'affichage des annonces

Mise en vente des véhicules : Par la suite, j'ai implémenté la possibilité de mettre en vente des véhicules. Pour cela, j'ai conçu un UserControl contenant un formulaire d'ajout avec tous les champs requis, y compris la possibilité de télécharger des images depuis l'ordinateur. L'application se chargera ensuite de les envoyer via une connexion FTP vers le serveur de stockage.

Figure 39 : Page de mise en vente

Modification et suppression d'annonces : Enfin, j'ai ajouté la fonctionnalité de modification et de suppression d'annonces. Pour la modification, j'ai adapté le même UserControl utilisé pour l'ajout en prenant en compte le contexte de modification. Quant à la suppression, elle nécessite une confirmation de l'utilisateur avant de désactiver l'annonce dans la base de données.

Dans l'ensemble, ce sprint s'est déroulé de manière satisfaisante, avec des durées de développement plus courtes que prévu pour la plupart des fonctionnalités. Je suis particulièrement satisfait d'avoir adopté l'utilisation des UserControls, qui ont permis une transition fluide entre les pages et ont simplifié la gestion du contenu de l'application.

4.1.3 Sprint 3

Création de la page de détail d'un véhicule : J'ai débuté ce sprint avec la création du UserControl qui contient la page de détail d'un véhicule. Cette page s'affiche lorsque l'utilisateur clique sur une annonce dans les résultats de recherche. Elle fournit des informations détaillées sur le véhicule sélectionné, ainsi que toutes les photos associées.

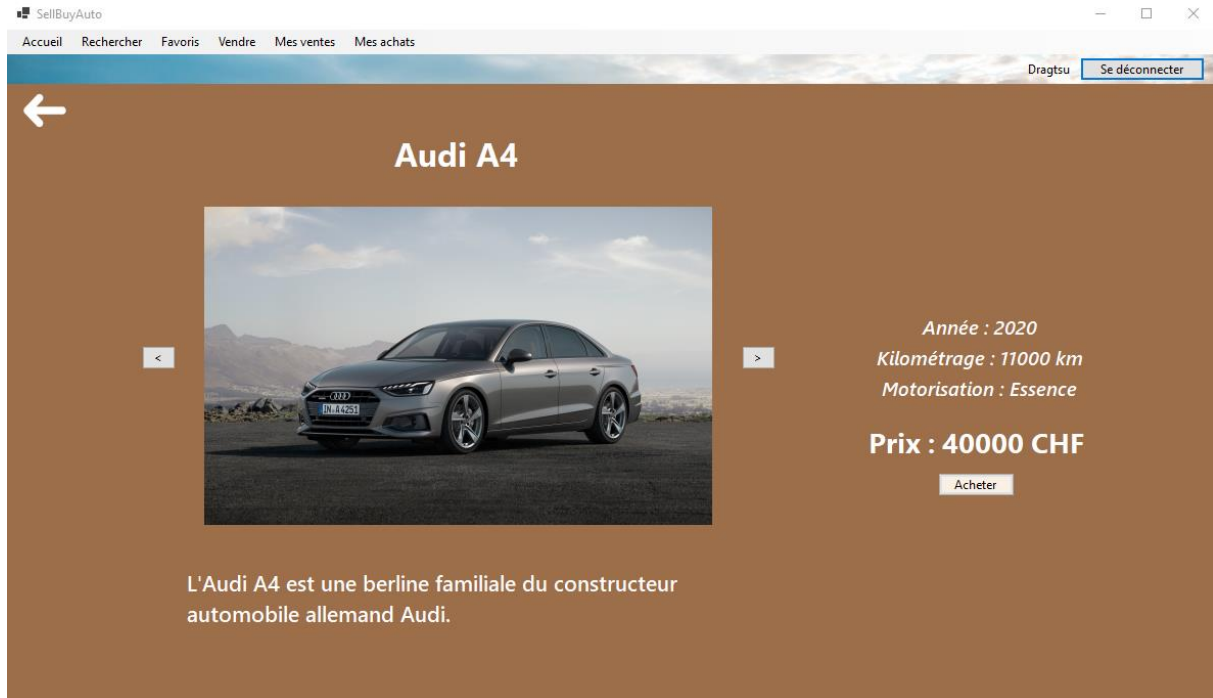


Figure 40 : Page de détail d'une annonce

Fonctionnalité d'achat de véhicules : Ensuite, j'ai ajouté la fonctionnalité permettant aux utilisateurs d'acheter un véhicule. Dans le UserControl de détail du véhicule, j'ai ajouté un bouton « Acheter ». Lorsque l'utilisateur clique sur ce bouton, un message apparaît avec l'adresse mail du vendeur pour faciliter le contact. De plus, l'adresse mail est automatiquement copiée dans le presse-papiers, ce qui permet à l'utilisateur de la coller directement.

Page « Mes favoris » : J'ai ensuite créé un autre UserControl contenant la page « Mes favoris » qui permet aux utilisateurs de voir les véhicules qu'ils ont ajoutés à leurs favoris. Pour cela, j'ai implémenté une étoile cliquable sur chaque annonce dans les résultats de recherche. Les utilisateurs peuvent cliquer sur cette étoile pour ajouter ou retirer un véhicule de leurs favoris. Les favoris peuvent également être gérés directement depuis la page « Mes favoris », où les utilisateurs peuvent les supprimer s'ils le souhaitent.

Page « Mes achats » : J'ai par la suite développé le UserControl avec la page « Mes achats » pour afficher les véhicules que les utilisateurs ont achetés. Cette page fonctionne comme un historique des achats. En cliquant sur une annonce dans cette page, les utilisateurs peuvent voir les détails du véhicule acheté, mais sans le bouton d'achat. Dans cette page, j'ai choisi de ne pas permettre la suppression des annonces, car elle sert à garder un historique des transactions.

Adaptation des pages « Mes favoris » et « Mes achats » : Pour les UserControls « Mes favoris » et « Mes achats », j'ai repris la structure du UserControl de la page « Mes ventes » et l'ai adaptée pour les nouvelles fonctionnalités. Cela m'a permis de gagner du temps tout en assurant une cohérence visuelle et fonctionnelle à travers l'application. Cette réutilisation des éléments de design et de code a également contribué à une expérience utilisateur plus uniforme. Pour ces deux pages, j'aurais pu utiliser le même UserControl que pour l'affichage de la page « Mes ventes » et ainsi éviter une duplication du code tout en gardant une cohérence dans l'interface et l'expérience utilisateur, mais j'ai estimé que cette méthode me prendrait beaucoup plus de temps à mettre en place. Ce point serait donc une amélioration possible du code.

Documentation : Enfin, j'ai consacré du temps à rédiger la documentation. J'ai détaillé les étapes de développement et les choix techniques effectués au cours du sprint.

En résumé, plusieurs fonctionnalités clé ont été implémentées. Elles permettent d'améliorer l'expérience utilisateur et la gestion des véhicules.

4.1.4 Sprint 4

Intégration des fonctionnalités pour les administrateurs : Dans ce sprint, j'ai intégré les fonctionnalités spécifiques aux administrateurs. Lorsqu'un administrateur est connecté, des boutons supplémentaires apparaissent sur les pages d'affichage des annonces après une recherche et sur la page de détail de l'annonce. Ces boutons permettent de bloquer une annonce ou de bloquer un utilisateur. Lorsqu'une annonce est bloquée, elle n'est plus visible depuis les recherches et devient uniquement visible par le vendeur dans la page « mes ventes » avec un message indiquant que l'annonce est bloquée. Enfin, Si un utilisateur est bloqué et qu'il essaye de se connecter, un message s'affiche indiquant que son compte est bloqué.

Finalisation de la documentation : Enfin, j'ai consacré du temps à finaliser la documentation. Cette dernière partie inclut la finalisation de la section réalisation, une conclusion avec un bilan personnel, un résumé du projet, et un glossaire.

Au cours de ce sprint, j'ai pu finir d'implémenter les principales fonctionnalités nécessaires au bon fonctionnement de l'application. En parallèle, j'ai également pu finaliser la documentation du projet.

4.2 Description des tests effectués

Tests	Action	Conditions	Réaction attendue	Réaction obtenue	Validé
S'enregistrer	Remplir le formulaire d'inscription avec les informations requises	Le nom d'utilisateur n'existe pas déjà dans la base de données	Un nouveau compte utilisateur sera créé dans la base de données et l'utilisateur sera redirigé vers la page d'accueil avec un message de succès	Un nouveau compte utilisateur est créé dans la base de données et l'utilisateur est redirigé vers la page d'accueil	Oui
		Le nom d'utilisateur existe déjà dans la base de données	Un message d'erreur indiquant que le nom d'utilisateur est déjà pris s'affichera	Un message d'erreur indiquant que le nom d'utilisateur est déjà pris s'affiche	Oui
Se connecter	Saisir le nom d'utilisateur et le mot de passe	Les informations de connexion sont correctes	L'utilisateur sera connecté à son compte et redirigé vers la page d'accueil	L'utilisateur est connecté à son compte et redirigé vers la page d'accueil	Oui
		Les informations de connexion sont incorrectes	Un message d'erreur indiquant que le login n'est pas correct s'affichera	Un message d'erreur indiquant que le login n'est pas correct s'affiche	Oui
Ajouter une annonce	Remplir les informations de l'annonce (marque, modèle, année, prix, etc.)	Toutes les informations obligatoires sont renseignées	L'annonce sera enregistrée dans la base de données et un message de succès s'affichera	L'annonce est enregistrée dans la base de données et un message de succès s'affiche	Oui
		Certaines informations obligatoires ne sont pas renseignées	Un message d'erreur indiquant les champs manquants s'affichera	Un message d'erreur indiquant des champs manquants s'affiche	Oui

Application C# de type plateforme de vente de voitures

Modifier une annonce	Sélectionner l'annonce à modifier et modifier les informations nécessaires	Toutes les informations obligatoires sont renseignées	Les informations de l'annonce seront mises à jour dans la base de données et un message de succès s'affichera	Les informations de l'annonce sont mises à jour dans la base de données et un message de succès s'affiche	Oui
		Certaines informations obligatoires ne sont pas renseignées	Un message d'erreur indiquant les champs manquants s'affichera	Un message d'erreur indiquant les champs manquants s'affiche	Oui
Supprimer une annonce	Sélectionner l'annonce à supprimer	Aucun utilisateur n'a mis l'annonce en favoris	L'annonce sera mise inactive dans la base de données et un message de confirmation s'affichera	L'annonce est mise inactive dans la base de données et un message de confirmation s'affiche	Oui
		Des utilisateurs ont mis l'annonce en favoris	Une confirmation sera demandée à l'utilisateur avant de supprimer l'annonce	Une confirmation est demandée à l'utilisateur avant de supprimer l'annonce	Oui
Mettre une annonce en favoris	Cliquer sur l'étoile à côté de l'annonce	L'utilisateur est connecté à son compte	L'annonce sera ajoutée aux favoris de l'utilisateur	L'annonce est ajoutée aux favoris de l'utilisateur	Oui
		L'utilisateur n'est pas connecté à son compte	Aucune étoile ne sera affichée	Aucune étoile n'est affichée	Oui
Consulter les annonces favorites	Cliquer sur le bouton « Favoris » dans le menu	L'utilisateur a des annonces en favoris	La liste des annonces favorites de l'utilisateur s'affichera	La liste des annonces favorites de l'utilisateur s'affiche	Oui
		L'utilisateur n'a pas d'annonces en favoris	Un message d'erreur indiquant qu'il n'y a pas d'annonces favorites s'affichera	Un message d'erreur indiquant qu'il n'y a pas d'annonces favorites s'affiche	Oui

Application C# de type plateforme de vente de voitures

Consulter ses ventes	Cliquer sur le bouton « Mes ventes » dans le menu	L'utilisateur a des annonces mises en vente	La liste des annonces mises en vente par l'utilisateur s'affichera	La liste des annonces mises en vente par l'utilisateur s'affiche	Oui
		L'utilisateur n'a pas d'annonces mises en vente	Un message d'erreur indiquant qu'il n'y a pas d'annonces mises en vente s'affichera	Un message d'erreur indiquant qu'il n'y a pas d'annonces mises en vente s'affiche	Oui
Consulter ses achats	Cliquer sur le bouton « Mes achats » dans le menu	L'utilisateur a des annonces achetées	La liste des annonces achetées par l'utilisateur s'affichera	La liste des annonces achetées par l'utilisateur s'affiche	Oui
		L'utilisateur n'a pas d'annonces achetées	Un message d'erreur indiquant qu'il n'y a pas d'annonces achetées s'affichera	Un message d'erreur indiquant qu'il n'y a pas d'annonces achetées s'affiche	Oui
Bloquer un utilisateur	Sélectionner l'utilisateur à bloquer	L'utilisateur est un administrateur	L'utilisateur sera mis comme bloqué dans la base de données et un message de confirmation s'affichera	L'utilisateur est mis comme bloqué dans la base de données et un message de confirmation s'affiche	Oui
Bloquer une annonce	Sélectionner l'annonce à bloquer	L'utilisateur est un administrateur	L'annonce sera mise comme bloquée dans la base de données et un message de confirmation s'affichera	L'annonce est mise comme bloquée dans la base de données et un message de confirmation s'affiche	Oui

Tableau 2: Test effectués

4.3 Test de l'Application par un Utilisateur Réel

Pour évaluer l'application, j'ai demandé à un utilisateur externe de tester l'application. J'ai fourni un tableau des cas d'utilisation pour que le testeur ait connaissance des fonctionnalités implémentées et puisse toutes les tester.

Dans l'ensemble, le test s'est bien passé. Le testeur a trouvé l'application intuitive et facile à utiliser. Les principales observations et résultats du test sont les suivants :

- **Intuitivité et facilité d'utilisation :** Le testeur a jugé l'application globalement intuitive. Les menus et les fonctionnalités étaient faciles à trouver et à utiliser, ce qui est un point positif pour l'expérience utilisateur.
- **Bugs identifiés et résolus :** Le testeur a identifié quelques bugs lors de l'utilisation de l'application. Ces bugs ont été pour la plupart résolus par la suite. Les retours du testeur ont été précieux pour améliorer la stabilité et la fiabilité de l'application.

4.4 Erreurs restantes

Dans le code que j'ai produit, les mots de passe, notamment ceux pour la connexion à la base de données et au serveur FTP, sont en clair et intégrés directement dans le code. Cela signifie que toute personne ayant accès au dépôt GitHub peut voir ces mots de passe. Je suis conscient que cette pratique est inappropriée pour un environnement de production en raison des risques de sécurité. Cependant, n'ayant pas encore acquis des connaissances approfondies sur des meilleures pratiques de sécurité en développement logiciel, j'ai adopté cette solution temporaire pour faciliter le développement de l'application. Je reconnais l'importance de sécuriser les informations sensibles, telles que les mots de passe, et j'ai commencé à explorer des solutions plus sécurisées. Parmi celles-ci, je considère le stockage des mots de passe dans des fichiers de configuration externes cryptés ou l'utilisation de services de gestion des secrets.

Lors du test effectué avec un utilisateur externe, une erreur a été identifiée après une utilisation prolongée de l'application. L'erreur rencontrée était un message « Out of memory ». Après réflexion, je pense que cette erreur est due au fait que les images téléchargées sont directement chargées en mémoire et ne se suppriment pas correctement.

J'avais prévu de gérer la suppression des images en utilisant la méthode « Dispose », mais il semble que cette méthode ne fonctionne pas comme prévu. Pour résoudre ce problème, il serait nécessaire de revoir tout le code pour s'assurer que les images sont effectivement supprimées lorsqu'elles ne sont plus utilisées.

5 Conclusion

Pour conclure, mon ressenti est globalement positif. Tous les objectifs initialement définis ont été atteints, ce qui constitue une réussite majeure. Je suis particulièrement satisfait de l'implémentation réussie de toutes les fonctionnalités prévues, ainsi que de la documentation réalisée.

En termes de comparaison entre ce qui était prévu et ce qui s'est réellement passé, l'implémentation des fonctionnalités a pris un peu moins de temps que prévu initialement. Cette efficacité m'a permis de prendre plus de temps à l'amélioration de ma documentation. J'ai pu détailler les différentes étapes de développement et fournir des explications claires sur les choix techniques effectués, ce qui à mon avis enrichit la qualité globale du projet.

En ce qui concerne les points positifs et négatifs, j'ai rencontré quelques difficultés lors de l'intégration de certaines fonctionnalités, mais j'ai finalement réussi à trouver des solutions satisfaisantes pour chacune des difficultés rencontrées. Le point positif majeur de ce projet est qu'il m'a permis de créer une application complète de manière autonome. Bien que la rédaction de la documentation ait représenté un défi, je suis satisfait du résultat final et de la manière dont j'ai pu surmonter cette difficulté. De plus, le retour positif de l'utilisateur externe lors des tests a confirmé la convivialité et l'utilité de l'application.

Pour améliorer encore l'application, plusieurs axes peuvent être explorés :

- **Critères de recherche supplémentaires** : Ajouter des critères comme la couleur du véhicule, l'état, et d'autres caractéristiques pertinentes pour affiner les recherches.
- **Types de véhicules variés** : Inclure la possibilité de vendre et d'acheter d'autres types de véhicules tels que des motos, des remorques, etc.
- **Optimisation des performances** : Améliorer l'optimisation du code pour que les pages se chargent plus rapidement et éviter le bug de mémoire après une longue utilisation.
- **Sécurité des données** : Revoir la gestion des mots de passe et des connexions aux serveurs pour éviter l'utilisation de mots de passe en dur dans le code, en mettant en place des pratiques de sécurité plus robustes.

Bilan personnel :

Pour exprimer mon ressenti sur ce projet, je suis pleinement satisfait des résultats obtenus. Tous les objectifs qui m'ont été fixés ont été atteints avec succès, ce qui constitue une réussite majeure pour moi. L'implémentation des fonctionnalités prévues s'est bien déroulée dans l'ensemble, renforçant ainsi mes compétences techniques et ma confiance en moi. Malgré quelques difficultés lors de l'intégration de certaines fonctionnalités, j'ai réussi à les surmonter avec succès. Initialement, la rédaction de la documentation représentait un vrai défi et j'étais anxieux à l'idée de le relever, mais finalement, je suis satisfait de la documentation que j'ai produite. Ce projet m'a offert une expérience précieuse et m'a permis de développer de nouvelles compétences que je pourrai appliquer dans de futurs projets.

6 Annexes

6.1 Résumé du rapport du TPI

Situation de départ :

Ce projet avait pour objectif de concevoir et développer une application de vente de véhicules en ligne. L'application devait permettre aux utilisateurs de publier des annonces, de rechercher des véhicules, et d'en acheter directement depuis la plateforme. De plus, il était important de prévoir des fonctionnalités spécifiques pour les administrateurs afin de gérer efficacement les annonces et les utilisateurs. La méthodologie agile a été adoptée pour assurer une progression structurée et flexible du projet.

Mise en œuvre :

La réalisation du projet s'est déroulée en plusieurs sprints, chacun avec des objectifs clairs et définis. Le premier sprint a été consacré à la mise en place de l'environnement de développement, à la création de la base de données et à la structure initiale de l'application. Les sprints suivants ont intégré des fonctionnalités telles que l'ajout d'annonces, la gestion des favoris, et la page de détail des véhicules. Le dernier sprint a introduit les fonctionnalités spécifiques aux administrateurs et a inclus des tests par un utilisateur externe pour évaluer l'application en conditions réelles. La documentation a été finalisée en parallèle.

Résultats :

Le projet a abouti à la création d'une application de vente de véhicules fonctionnelle et intuitive. Les utilisateurs peuvent désormais publier des annonces, rechercher des véhicules, ajouter des véhicules en favoris et effectuer des achats. Les administrateurs disposent de fonctions pour gérer les annonces et les utilisateurs. Les tests effectués par un utilisateur externe ont confirmé la convivialité de l'application, bien qu'ils aient également révélé quelques bugs, dont certains ont été corrigés. Des pistes d'amélioration ont été identifiées, telles que l'optimisation des performances et l'ajout de nouveaux critères de recherche et types de véhicules.

6.2 Glossaire

C

Contrôle

Le contrôle est un composant graphique interactif qui est utilisé dans les projets Windows form 29, 30

D

Dispose

La méthode Dispose en C# est utilisée pour libérer les ressources non utilisées d'un objet, assurant ainsi une gestion efficace de la mémoire 37

F

FlowLayoutPanel

Le FlowLayoutPanel est un contrôle qui organise dynamiquement les contrôles qu'il contient en fonction de l'espace disponible et gère automatiquement le défilement lorsque nécessaire 30

FTP

Le FTP (File Transfer Protocol) est un protocole permettant le transfert de fichiers entre un client et un serveur sur un réseau, comme Internet. Il facilite le téléchargement et le téléversement de fichiers 23, 31, 37, 41

S

Sprint

En méthodologie agile, un sprint est une période de temps définie, pendant laquelle une équipe se concentre sur la réalisation d'un ensemble spécifique de tâches, aboutissant à un livrable potentiellement fonctionnel 4, 20, 21, 22, 27, 28, 31, 32, 33, 39

Stories

En méthodologie agile, une "story" (ou "user story") représente une description simple et concise d'une fonctionnalité ou d'un besoin du point de vue de l'utilisateur. Elle sert de base à la planification et à la réalisation des tâches lors des sprints 27

U

UserControl

Un UserControl est un élément qui appartient au Windows Form et qui permet de créer un contrôle personnalisé 29, 30, 31, 32, 33

W

Windows Form

Un Windows Form est un type de projet sur Visual Studio qui permet de créer des applications graphiques 23, 28, 29

6.3 Sources – Bibliographie

1. ChatGPT

Description : Site officiel de ChatGPT, une intelligence artificielle développée par OpenAI.
Disponible sur : <https://chat.openai.com/>

2. Fermer main form

Description : Comment fermer un formulaire principal dans une application Windows Forms.

Disponible sur : <https://stackoverflow.com/questions/17820995/close-main-form>

3. Télécharger fichiers avec une connexion FTP

Description : Tutoriel sur le téléchargement de fichiers en utilisant une connexion FTP avec .NET.

Disponible sur : <https://learn.microsoft.com/en-us/dotnet/framework/network-programming/how-to-download-files-with-ftp>

4. Tutoriel vidéo sur le téléchargement de fichiers avec une connexion FTP

Description : Vidéo explicative sur YouTube concernant le téléchargement de fichiers via FTP.

Disponible sur : <https://www.youtube.com/watch?v=5d-AE21Zjog>

5. Run méthode en arrière-plan

Description : Exécution de méthodes asynchrones sur un thread en arrière-plan en C#.

Disponible sur : <https://stackoverflow.com/questions/20304258/run-async-method-on-a-background-thread>

6. Supprimer variable de la mémoire

Description : Comment supprimer définitivement une variable de la mémoire en C#.

Disponible sur : <https://security.stackexchange.com/questions/212917/how-to-delete-a-c-variable-value-permanently-from-memory>

7. Ajouter des contrôles dans un FlowLayoutPanel

Description : Tutoriel vidéo sur l'ajout de contrôles dans un FlowLayoutPanel en C#.

Disponible sur : <https://www.youtube.com/watch?v=VeapnO7b2gI>

8. Limiter les filtres dans OpenFileDialog

Description : Comment définir les filtres pour un OpenFileDialog afin de permettre uniquement les formats d'image courants.

Disponible sur : <https://stackoverflow.com/questions/2069048/setting-the-filter-to-an-openfiledialog-to-allow-the-typical-image-formats>

9. Message de confirmation avant de supprimer

Description : Tutoriel vidéo sur l'affichage d'un message de confirmation avant de supprimer un élément.

Disponible sur : <https://www.youtube.com/watch?v=rjETrJG0a0w>

10. Stop méthode async

Description : Méthode pour arrêter une méthode asynchrone en cours d'exécution en C#.

Disponible sur : <https://stackoverflow.com/questions/15614991/simply-stop-an-async-method>

11. Utilisation de CancellationToken pour arrêter une méthode asynchrone

Description : Documentation sur l'utilisation de CancellationToken pour gérer l'annulation des tâches asynchrones.

Disponible sur : <https://learn.microsoft.com/en-us/dotnet/api/system.threading.cancellationtoken.throwifcancellationrequested?view=net-8.0>

12. Lancer l'événement click avec le code

Description : Comment invoquer par programmation l'événement Click d'un bouton en C#.

Disponible sur : <https://bytes.com/topic/c-sharp/472035-how-programmatically-invoke-click-event-button>

13. Copier du texte dans le presse-papiers

Description : Comment copier des données dans le presse-papiers en utilisant C#.

Disponible sur : <https://stackoverflow.com/questions/3546016/how-to-copy-data-to-clipboard-in-c-sharp>

14. Attendre qu'une tâche soit terminée

Description : Documentation sur la méthode Task.Wait pour attendre la fin d'une tâche en C#.

Disponible sur : <https://learn.microsoft.com/en-us/dotnet/api/system.threading.tasks.task.wait?view=net-8.0>

6.4 Journal de Travail

Date	Sprint	Activité	Heures	Remarques
30.04.2024	1	1.1 - Lecture du cahier des charges	0:30	Lecture du cahier des charges et entretien avec le premier expert. Avis personnel : Je me sens plutôt confiant concernant le travail à réaliser
30.04.2024	1	2.1 - Mise en place de l'environnement	1:30	Création de l'IceScrum avec les sprints et les stories
30.04.2024	1	1.3 - Planification	2:00	Création de la planification initiale
01.05.2024	1	1.3 - Planification	1:00	Finalisation de la planification initiale
01.05.2024	1	4.2 - Rédaction de la documentation	1:00	Mise en place de la documentation
01.05.2024	1	2.1 - Mise en place de l'environnement	1:00	Création du dépôt sur GitHub et du projet Visual Studio
01.05.2024	1	1.4 - Maquettes	0:40	Début de la réalisation des maquettes
01.05.2024	1	4.1 - Journal de travail	0:20	Mise en place du journal de travail
02.05.2024	1	1.4 - Maquettes	1:20	Terminer la réalisation des maquettes
02.05.2024	1	1.6 - MCD/MLD	1:50	Création du MCD
02.05.2024	1	1.6 - MCD/MLD	1:40	Création du MLD
02.05.2024	1	2.2 - Base de données	1:15	Création de la base de données sur swisscenter et début de la génération des données
02.05.2024	1	4.1 - Journal de travail	0:15	Remplir le journal de travail + envoi de l'email semi-hebdomadaire
03.05.2024	1	1.7 - Daily meeting	0:30	Revue du dépôt GitHub, le fait de séparer les PDF c'est très bien. Revue des maquettes -> ok Revue de la planification initiale, bonne idée d'avoir séparé chaque fonctionnalité. Revue de la documentation, le fait de reprendre les objectifs du CDC, c'est bien. Revue de l'IceScrum, les tâches sont bien respectées et mis à jour. Revue des données dans le fichier insert SQL
03.05.2024	1	2.2 - Base de données	1:55	Recherche d'images pour chaque annonce de la base de données

Application C# de type plateforme de vente de voitures

Date	Sprint	Activité	Heures	Remarques
03.05.2024	1	1.5 - Diagrammes	1:00	Création du diagramme de flux
03.05.2024	1	1.4 - Maquettes	0:05	Correction des maquettes, il manquait la maquette pour la page des favoris
03.05.2024	1	1.5 - Diagrammes	0:55	Début de la création du diagramme de classe
03.05.2024	1	4.1 - Journal de travail	0:20	Remplir le journal de travail + envoi de l'email semi-hebdomadaire
06.05.2024	1	1.5 - Diagrammes	0:45	Finaliser le diagramme de classe
06.05.2024	1	1.6 - MCD/MLD	0:10	Correction du MCD et du MLD, ajout de l'entité modèles
06.05.2024	1	2.2 - Base de données	0:30	Correction de la base de données suite aux modifications dans le MCD/MLD
06.05.2024	1	4.2 - Rédaction de la documentation	2:35	J'ai ajouté les diagrammes ainsi que le MCD et le MLD. J'ai également créé les use cases et scénarios
06.05.2024	1	1.8 - Sprint review	0:20	Revue de toutes les stories du premier sprint. On a refait tous les tests ensemble. Le premier sprint est validé.
06.05.2024	1	4.2 - Rédaction de la documentation	1:15	Ajout des maquettes dans la documentation
06.05.2024	1	1.3 - Planification	0:35	Ajout des stories du sprint 2 sur IceScrum
06.05.2024	1	4.1 - Journal de travail	0:10	Remplir le journal de travail
07.05.2024	2	4.2 - Rédaction de la documentation	0:30	Compléter la partie réalisation avec le sprint 1
07.05.2024	2	2.5 - Rechercher des véhicules selon filtres	0:30	Création de la page d'accueil
07.05.2024	2	2.3 - Login/Register	1:15	Création des pages de login et de register + coder les 2 fonctionnalités
07.05.2024	2	2.5 - Rechercher des véhicules selon filtres	1:35	Création du Form pour la recherche avancée mais j'ai eu une nouvelle idée. Au lieu de créer plein de formulaire et d'avoir un changement de fenêtre constant, je vais utiliser des UserControl afin d'afficher le contenu souhaité, ainsi la transition entre les pages se fera de manière plus fluide Avis Personnel : Je suis un peu inquiet parce que je ne sais pas si la nouvelle idée va fonctionner ou non

Application C# de type plateforme de vente de voitures

Date	Sprint	Activité	Heures	Remarques
07.05.2024	2	4.1 - Journal de travail	0:10	Remplir le journal de travail
08.05.2024	2	2.5 - Rechercher des véhicules selon filtres	0:30	Mise en place de la nouvelle idée avec les UserControls Avis Personnel : Je suis plutôt content et satisfait que mon idée fonctionne correctement
08.05.2024	2	2.5 - Rechercher des véhicules selon filtres	2:40	Création des requêtes SQL
08.05.2024	2	2.4 - Visualisation des véhicules en ventes	0:35	Création de la page de visualisation des véhicules rechercher
08.05.2024	2	4.1 - Journal de travail	0:15	Remplir le journal de travail + envoi de l'email semi-hebdomadaire
13.05.2024	2	2.4 - Visualisation des véhicules en ventes	2:45	Création du UserControl qui permet d'afficher les annonces + création de la requête FTP + Ajout de commentaires
13.05.2024	2	2.6 - Mettre en vente un véhicule	0:30	Création du UserControl contenant le formulaire pour ajouter/modifier une annonce
13.05.2024	2	1.7 - Daily meeting	0:20	Discussion sur le travail qui a déjà été réalisé
13.05.2024	2	2.6 - Mettre en vente un véhicule	2:30	Création des requêtes SQL et FTP
13.05.2024	2	4.1 - Journal de travail	0:15	Remplir le journal de travail
14.05.2024	2	2.6 - Mettre en vente un véhicule	0:40	Terminer les requêtes qui permettent d'ajouter les véhicules
14.05.2024	2	2.6 - Mettre en vente un véhicule	0:20	Création de la page mes ventes
14.05.2024	2	2.7 - Modifier/Supprimer ses annonces	1:15	Coder la possibilité de modifier une annonce
14.05.2024	2	2.7 - Modifier/Supprimer ses annonces	0:45	Coder la possibilité de supprimer une annonce Avis personnel : Je suis soulagé parce que j'ai réussi à finir les fonctionnalités du sprint 2 en avance
14.05.2024	2	3.1 - Tests unitaires	0:45	Tests de chacune des fonctionnalités, créer jusqu'à maintenant
14.05.2024	2	4.1 - Journal de travail	0:15	Remplir le journal de travail
15.05.2024	2	3.2 - Tests d'intégrations	1:45	Tester que les fonctionnalités ne rentrent pas en conflits les unes avec les autres

Application C# de type plateforme de vente de voitures

Date	Sprint	Activité	Heures	Remarques
15.05.2024	2	4.2 - Rédaction de la documentation	1:00	Rédaction de la partie réalisation du sprint 2
15.05.2024	2	1.2 - Recherches	1:00	Recherche sur comment créer un fichier exe afin de transmettre l'application
15.05.2024	2	4.1 - Journal de travail	0:15	Remplir le journal de travail + envoi de l'email semi-hebdomadaire
16.05.2024	2	1.7 - Daily meeting	0:30	Entretien avec l'expert 2, revue du projet dans sa globalité, dans l'ensemble tout se passe bien. Pour la documentation, la partie réalisation, il faut demander à M. Benzonana si cela lui convient. Nous avons également discuté sur le fait que les mots de passe sont écrits en dur dans le code. Pour l'envoi de l'exécutable, il faudrait un lien pour télécharger le dossier zip. M. Melly m'a également montré comment créer un fichier exécutable afin de transmettre l'application plus facilement avec un seul fichier.
16.05.2024	2	3.2 - Tests d'intégrations	1:00	Test de l'application afin de trouver d'autres bugs ou des choses à améliorer.
16.05.2024	2	2.4 - Visualisation des véhicules en ventes	1:30	Correction d'un bug qui faisait que les images continuaient de charger en arrière-plan même quand on changeait de page.
16.05.2024	2	1.8 - Sprint review	0:50	Revue de toutes les stories du deuxième sprint. On a refait tous les tests ensemble. Le deuxième sprint est validé. J'ai demandé au chef de projet si la partie de la réalisation divisée par sprint lui convenait et il m'a répondu que oui, étant donné que je travaille avec la méthode agile, il trouve cette façon de faire cohérente.
16.05.2024	2	1.3 - Planification	0:40	Ajout des stories du sprint 3 sur IceScrum
16.05.2024	2	4.2 - Rédaction de la documentation	1:35	Ajout des stories du sprint 3 et ajout de titres pour la partie réalisation.
16.05.2024	2	4.1 - Journal de travail	0:15	Remplir le journal de travail
17.05.2024	3	2.8 - Acheter un véhicule	1:30	Création du UserControl qui affiche le détail des annonces + coder la fonctionnalité
17.05.2024	3	2.8 - Acheter un véhicule	2:10	Coder la fonctionnalité de vente pour le vendeur, pour qu'il puisse choisir l'acheteur

Application C# de type plateforme de vente de voitures

Date	Sprint	Activité	Heures	Remarques
17.05.2024	3	3.1 - Tests unitaires	0:30	Différent test pour voir si la nouvelle fonctionnalité fonctionne correctement
17.05.2024	3	1.2 - Recherches	0:20	Recherche sur comment copier dans le presse-papiers et comment lancer un event click depuis le code
17.05.2024	3	4.1 - Journal de travail	0:15	Remplir le journal de travail + envoi de l'email semi-hebdomadaire
21.05.2024	3	2.9 - Mettre en favoris	1:00	Création de la page mes favoris + supprimer favoris
21.05.2024	3	2.9 - Mettre en favoris	1:00	Coder la possibilité de mettre en favoris depuis la page d'affichage des recherches
21.05.2024	3	2.10 - Historique des ventes/achats	1:00	Création de la page mes achats avec les requêtes nécessaires Avis personnel : Je suis soulagé parce que j'ai réussi à finir les fonctionnalités du sprint 3 en avance
21.05.2024	3	3.2 - Tests d'intégrations	0:45	Tests de l'application en général
21.05.2024	3	4.1 - Journal de travail	0:15	Remplir le journal de travail
22.05.2024	3	4.2 - Rédaction de la documentation	3:20	Ajouter la partie réalisation du sprint 3 + Ajout du tableau avec les tests effectués + ajout des stories du sprint 4
22.05.2024	3	1.3 - Planification	0:25	Ajout des stories du sprint 4 sur IceScrum
22.05.2024	3	4.1 - Journal de travail	0:15	Remplir le journal de travail + envoi de l'email semi-hebdomadaire
23.05.2024	3	3.2 - Tests d'intégrations	2:00	Test général de l'application + Test sur VM windows 10 et sur Windows 11. L'apparition de la fenêtre d'enregistrement se fait maintenant à la place de celle de login + Amélioration du design
23.05.2024	3	4.2 - Rédaction de la documentation	1:00	Mise en forme des sources + Ajout d'images dans la partie réalisation
23.05.2024	3	1.8 - Sprint review	1:00	Revue de tous les tests sur IceScrum et discussion sur un bug lié à la fenêtre de login
23.05.2024	3	2.4 - Visualisation des véhicules en ventes	2:05	Modification de l'aspect général de l'application (Design)
23.05.2024	3	4.1 - Journal de travail	0:15	Remplir le journal de travail
24.05.2024	4	2.3 - Login/Register	1:20	Création des UserControls pour le login et l'enregistrement

Application C# de type plateforme de vente de voitures

Date	Sprint	Activité	Heures	Remarques
24.05.2024	4	4.2 - Rédaction de la documentation	0:30	Ajout des nouvelles images dans la réalisation
24.05.2024	4	2.11 - Bloquer une annonce (Admin)	1:40	Résolution d'un bug qui faisait planter le programme lorsque qu'on effectuait des recherches en étant admin + Ajout du bouton Bloquer annonce ainsi que les requête SQL
24.05.2024	4	2.12 - Bloquer un utilisateur (Admin)	1:00	Ajout du bouton Bloquer un utilisateur ainsi que les requête SQL
24.05.2024	4	4.1 - Journal de travail	0:15	Remplir le journal de travail + envoie de l'email semi-hebdomadaire
27.05.2024	4	2.7 - Modifier/Supprimer ses annonces	0:20	Correction d'un bug qui empêchait de modifier et de supprimer des annonces
27.05.2024	4	1.5 - Diagrammes	0:10	Mise à jour du diagramme de classes
27.05.2024	4	1.4 - Maquettes	0:15	Correction des maquettes
27.05.2024	4	1.6 - MCD/MLD	0:20	Changer la version du MCD et du MLD
27.05.2024	4	3.2 - Tests d'intégrations	2:00	Test de l'application afin de trouver d'autres bugs ou des choses à améliorer.
27.05.2024	4	4.2 - Rédaction de la documentation	3:00	Rédaction de la partie réalisation du sprint 4 + la conclusion + le résumer du projet
27.05.2024	4	4.1 - Journal de travail	0:15	Remplir le journal de travail
28.05.2024	4	3.1 - Tests unitaires	0:15	Test de tous les cas d'utilisation
28.05.2024	4	4.2 - Rédaction de la documentation	3:30	Relecture de la documentation et amélioration
28.05.2024	4	4.1 - Journal de travail	0:15	Remplir le journal de travail
29.05.2024	4	4.2 - Rédaction de la documentation	1:45	Finalisation de la documentation + envoie du mail Final Avis personnel : Je suis content que le projet soit fini

Tableau 3 : Journal de travail

6.5 Planification finale

Projet

Application C# de type plateforme de vente de voiture

Planification

Total		30.04.24	07.05.24	17.05.24	24.05.24
Prévu	90 h 00	25 h 25	28 h 40	19 h 05	16 h 50
Da Silva	90 h 00	25 h 25	28 h 40	19 h 05	16 h 50

Sprint

1

2

3

4

1 Analyse

18 h 00

19 h 10

1.1 - Lecture du cahier des charges	Prévu	0 h 30				0 h 30	
	Da Silva	0 h 30	0 h 00	0 h 00	0 h 00	0 h 30	
	1.2 - Recherches	Prévu		0 h 40	0 h 40		1 h 20
		Da Silva	0 h 00	1 h 00	0 h 20	0 h 00	1 h 20
	1.3 - Planification	Prévu	3 h 25				3 h 25
		Da Silva	3 h 35	0 h 40	0 h 25	0 h 00	4 h 40
	1.4 - Maquettes	Prévu	4 h 40				4 h 40
		Da Silva	2 h 05	0 h 00	0 h 00	0 h 15	2 h 20
	1.5 - Diagrammes	Prévu	3 h 30				3 h 30
		Da Silva	2 h 40	0 h 00	0 h 00	0 h 10	2 h 50
	1.6 - MCD/MLD	Prévu	3 h 30				3 h 30
		Da Silva	3 h 40	0 h 00	0 h 00	0 h 20	4 h 00
	1.7 - Daily meeting	Prévu					
		Da Silva	0 h 30	0 h 50	0 h 00	0 h 00	1 h 20
	1.8 - Sprint review	Prévu	0 h 20	0 h 25	0 h 20		1 h 05
		Da Silva	0 h 20	0 h 50	1 h 00	0 h 00	2 h 10

2 Implémentation

40 h 30

36 h 35

2.1 - Mise en place de l'environnement	Prévu	3 h 30				3 h 30	
	Da Silva	2 h 30	0 h 00	0 h 00	0 h 00	2 h 30	
	2.2 - Base de données	Prévu	3 h 00				3 h 00
		Da Silva	3 h 40	0 h 00	0 h 00	0 h 00	3 h 40
	2.3 - Login/Register	Prévu		4 h 00			4 h 00
		Da Silva	0 h 00	1 h 15	0 h 00	1 h 20	2 h 35
	2.4 - Visualisation des véhicules en ventes	Prévu		5 h 00			5 h 00
		Da Silva	0 h 00	4 h 50	2 h 05	0 h 00	6 h 55
	2.5 - Rechercher des véhicules selon filtres	Prévu		4 h 00			4 h 00
		Da Silva	0 h 00	5 h 15	0 h 00	0 h 00	5 h 15
	2.6 - Mettre en vente un véhicule	Prévu		4 h 00			4 h 00
		Da Silva	0 h 00	4 h 00	0 h 00	0 h 00	4 h 00
	2.7 - Modifier/Supprimer ses annonces	Prévu		2 h 00			2 h 00
		Da Silva	0 h 00	2 h 00	0 h 00	0 h 20	2 h 20
	2.8 - Acheter un véhicule	Prévu			2 h 30		2 h 30
		Da Silva	0 h 00	0 h 00	3 h 40	0 h 00	3 h 40
	2.9 - Mettre en favoris	Prévu			3 h 00		3 h 00
		Da Silva	0 h 00	0 h 00	2 h 00	0 h 00	2 h 00
	2.10 - Historique des ventes/achats	Prévu			4 h 30		4 h 30
		Da Silva	0 h 00	0 h 00	1 h 00	0 h 00	1 h 00
	2.11 - Bloquer une annonce (Admin)	Prévu				2 h 30	2 h 30
		Da Silva	0 h 00	0 h 00	0 h 00	1 h 40	1 h 40
	2.12 - Bloquer un utilisateur (Admin)	Prévu				2 h 30	2 h 30
		Da Silva	0 h 00	0 h 00	0 h 00	1 h 00	1 h 00

3 Tests

9 h 00

9 h 00

3.1 - Tests unitaires	Prévu		1 h 00	1 h 15	0 h 30	2 h 45	
	Da Silva	0 h 00	0 h 45	0 h 30	0 h 15	1 h 30	
	3.2 - Tests d'intégrations	Prévu		1 h 35	2 h 00	1 h 40	5 h 15
		Da Silva	0 h 00	2 h 45	2 h 45	2 h 00	7 h 30
	3.3 - Test d'acceptation	Prévu				1 h 00	1 h 00
		Da Silva	0 h 00	0 h 00	0 h 00	0 h 00	

4 Documentation

22 h 30

25 h 15

4.1 - Journal de travail	Prévu	2 h 00	1 h 00	0 h 40	0 h 40	4 h 20
	Da Silva	1 h 05	1 h 25	1 h 00	0 h 45	4 h 15
4.2 - Rédaction de la documentation	Prévu	1 h 00	5 h 00	4 h 10	8 h 00	18 h 10
	Da Silva	4 h 50	3 h 05	4 h 20	8 h 45	21 h 00

Figure 41 : Planification finale