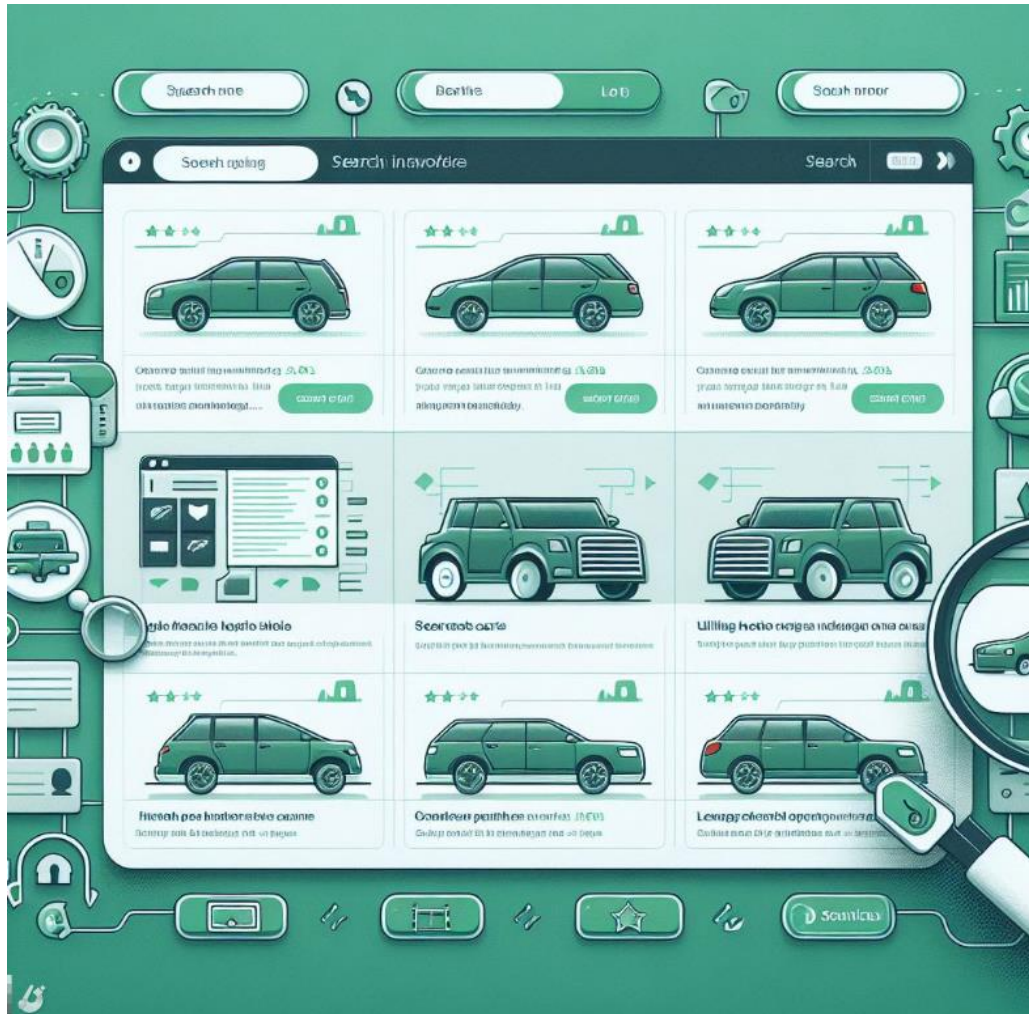


# Application C# de type plateforme de vente de voitures



Maikol Correia Da Silva  
Maikol.Correia@eduvaud.ch

## Table des matières

1	Introduction.....	3
1.1	Cadre, description et motivation.....	3
1.2	Organisation .....	3
1.3	Objectifs.....	3
1.4	Planification initiale .....	4
2	Analyse.....	6
2.1	Cahier des charges détaillé.....	6
2.1.1	MCD.....	6
2.1.2	Maquettes .....	7
2.1.3	Use Cases et scénarios .....	19
2.2	Stratégie de test .....	20
2.3	Planification .....	21
2.3.1	Sprint 1 .....	21
2.3.2	Sprint 2 .....	21
2.3.3	Sprint 3 .....	22
2.3.4	Sprint 4 .....	22
3	Conception.....	23
3.1	Logiciels.....	23
3.2	Choix techniques .....	23
3.3	MLD.....	24
3.4	Diagramme de classes .....	25
3.5	Diagramme de flux .....	26
4	Réalisation .....	27
4.1	Dossier de réalisation .....	27
4.1.1	Sprint 1 .....	27
4.1.2	Sprint 2 .....	28
4.1.3	Sprint 3 .....	29
4.1.4	Sprint 4 .....	30
4.2	Description des tests effectués .....	31
4.3	Test de l'Application par un Utilisateur Réel .....	34
4.4	Erreurs restantes .....	34
5	Conclusion .....	34
6	Annexes .....	34
6.1	Résumé du rapport du Pré-TPI .....	34
6.2	Glossaire .....	34
6.3	Sources – Bibliographie .....	34
6.4	Journal de Travail.....	35
6.5	Code du script de génération de données .....	35

## 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 voiture d'occasion, tout en mettant l'accent sur la gestion efficace d'une base de données. En tant que passionné de voitures, je suis enthousiasmé par l'opportunité de développer un logiciel qui porte sur ce sujet.

### 1.2 Organisation

Élève 1 : Correia Da Silva, Maikol, [maikol.correia@eduvaud.ch](mailto:maikol.correia@eduvaud.ch)

Responsable de projet : Benzonana, Pascal, [pascal.benzonana@eduvaud.ch](mailto:pascal.benzonana@eduvaud.ch)

Expert 1 : Montemayor, Ernesto, [ernesto@bati-technologie.ch](mailto:ernesto@bati-technologie.ch)

Expert 2 : Melly, Jonathan, [Jonathan.melly@eduvaud.ch](mailto: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 critères de 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 favoris
- 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 le 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	
2.3 - Login/Register	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	
	Prévu			1 h 00		1 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	
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

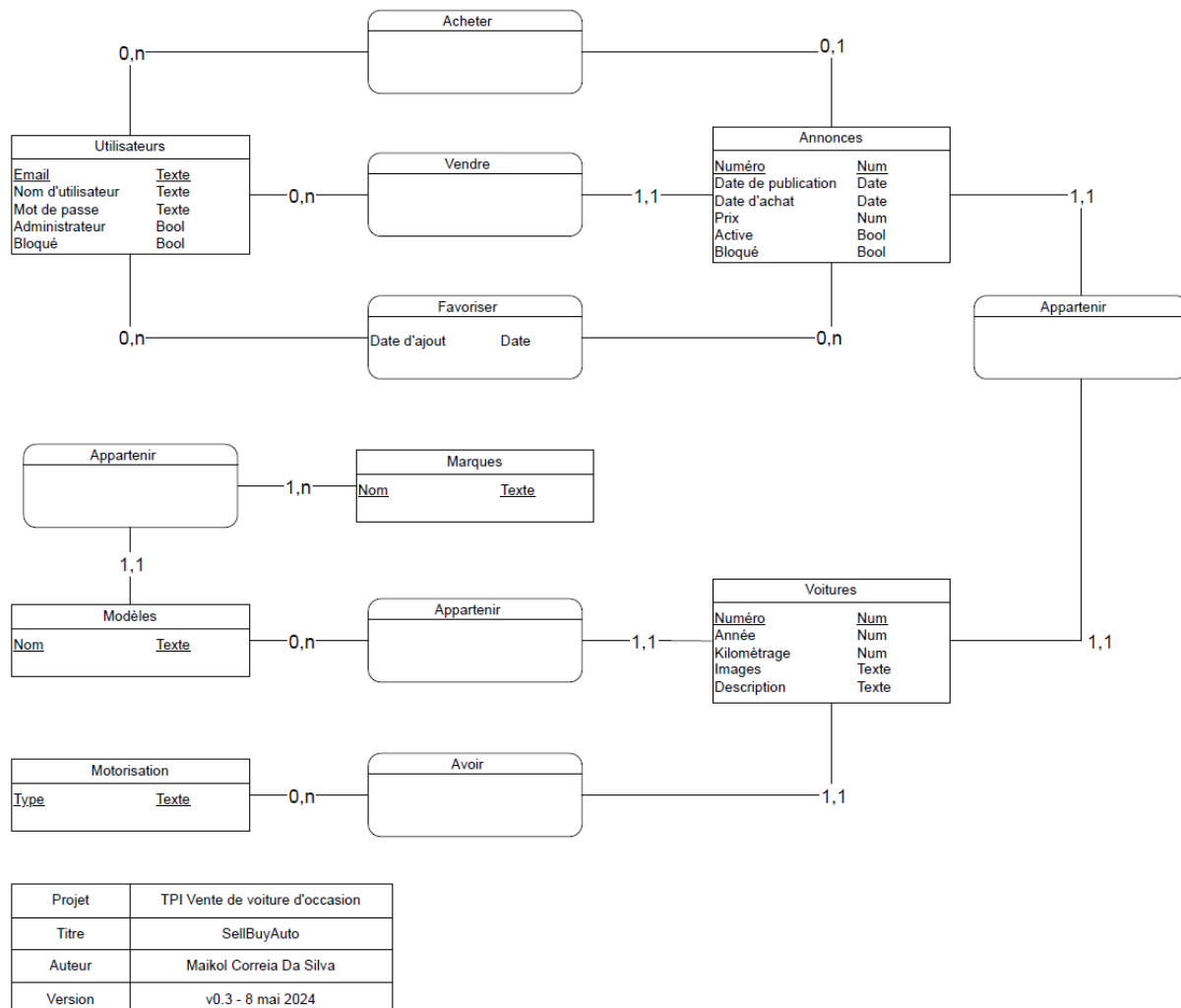


Figure 2 : MCD

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 à également 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

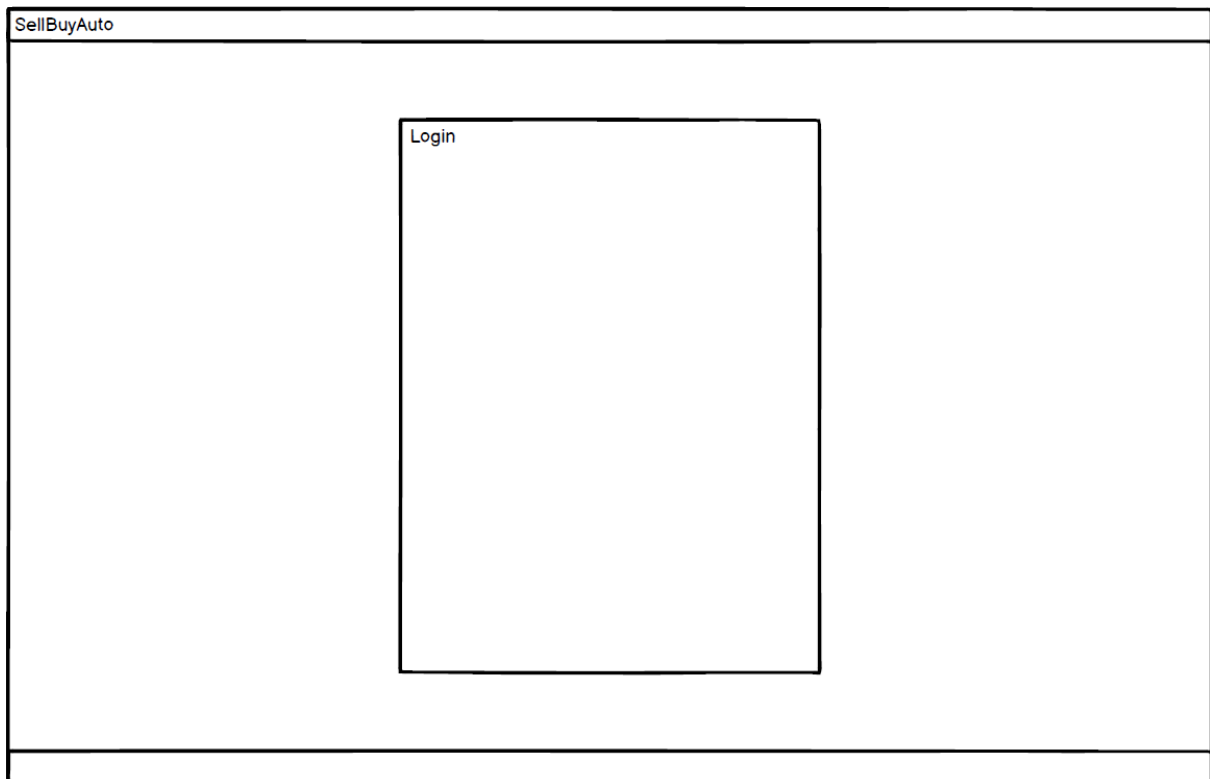


Figure 3 : LoginPage\_Zonning



Figure 4 : LoginPage\_Wireframe

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

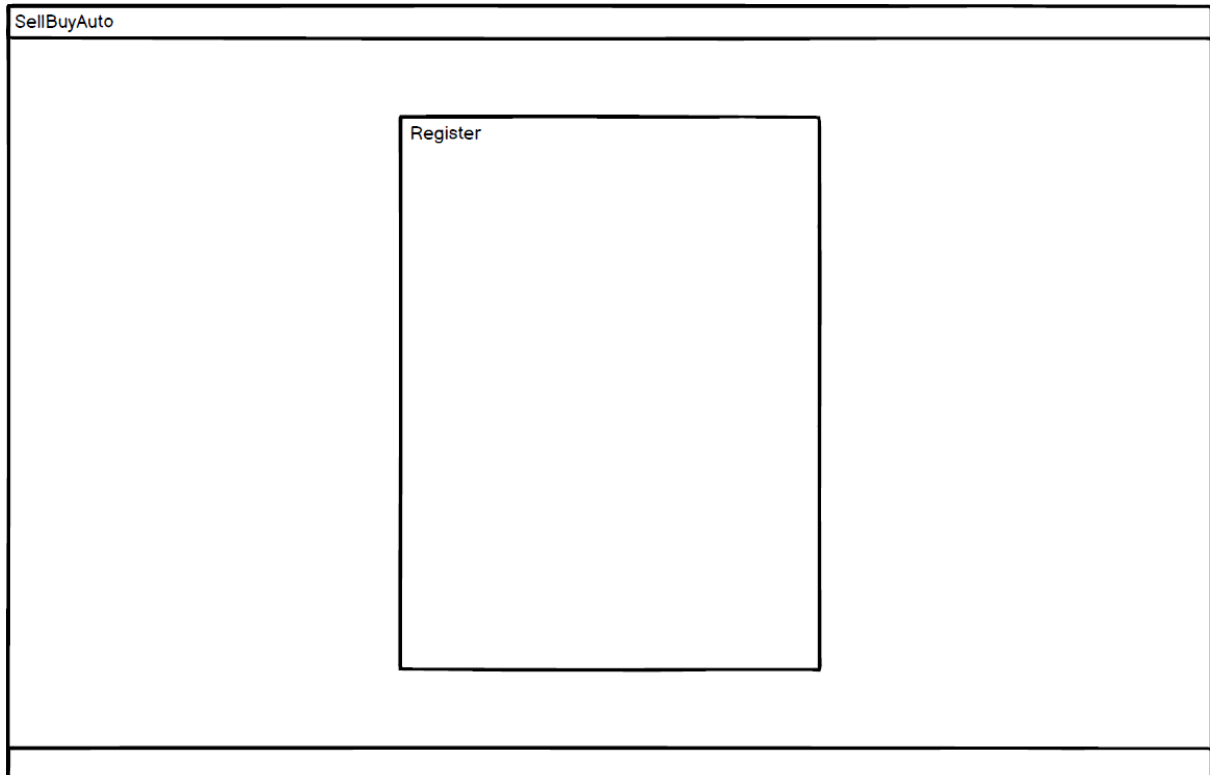


Figure 5 : RegisterPage\_Zonning



Figure 6 : RegisterPage\_Wireframe

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



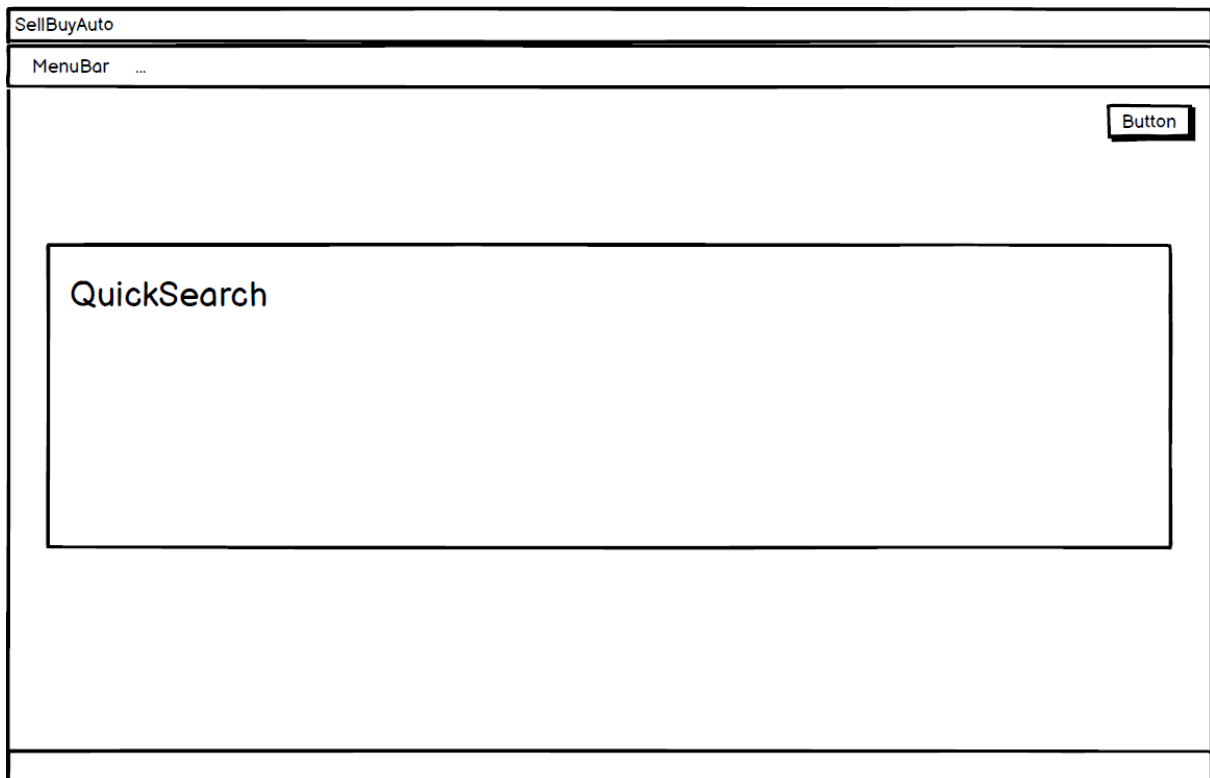


Figure 7 : HomePageWithoutLogin\_Zonning

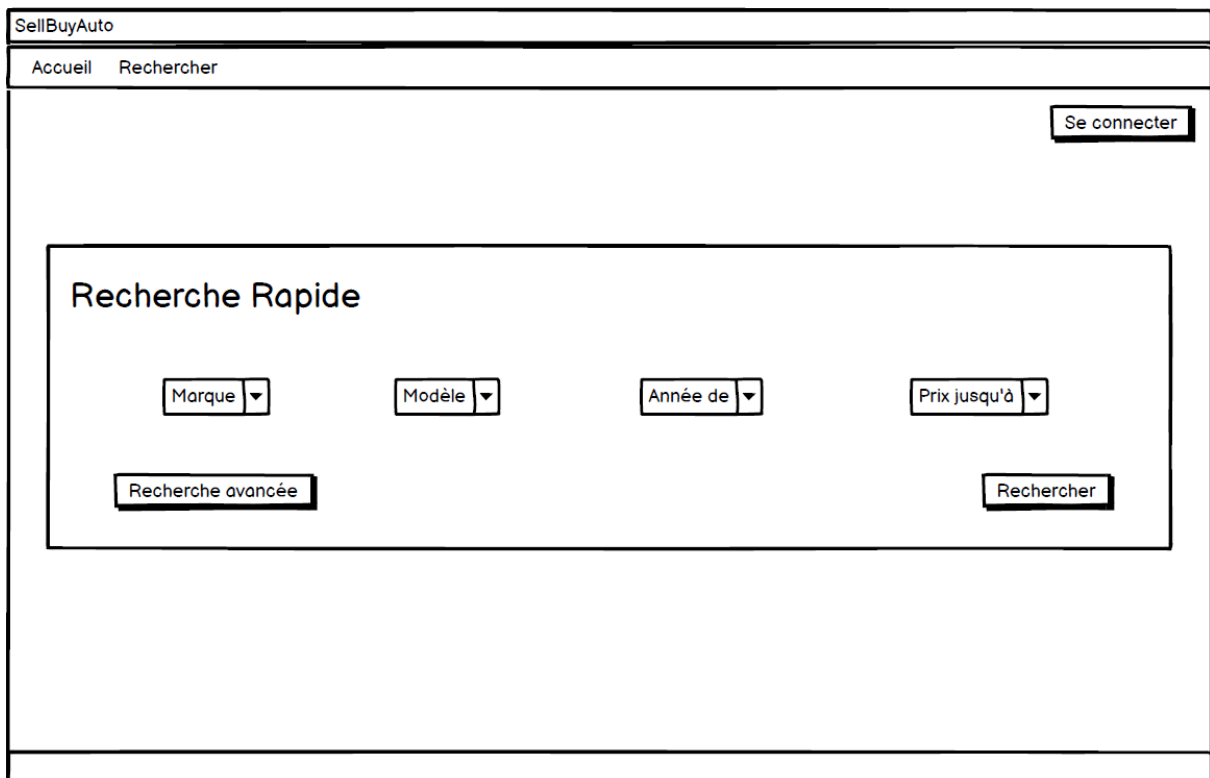


Figure 8 : HomePageWithoutLogin\_Wireframe

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.

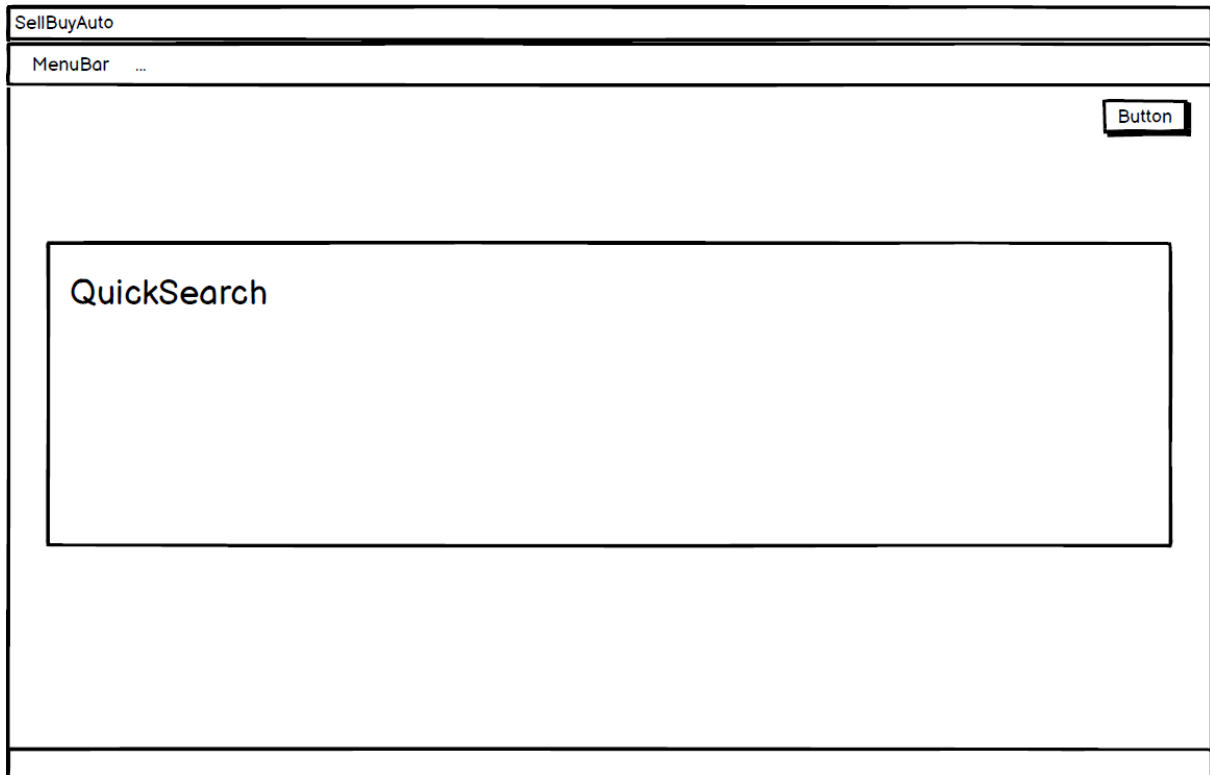


Figure 9 : HomePageWithLogin\_Zonning

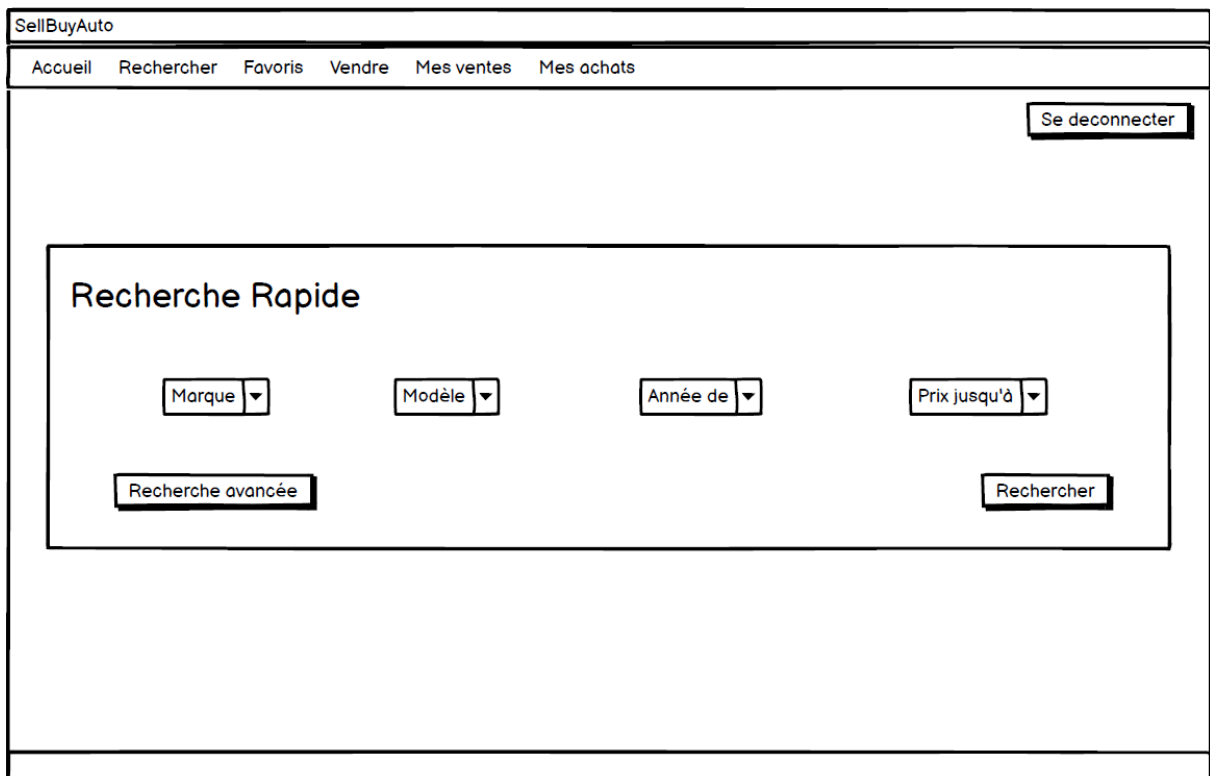


Figure 10 : HomePageWithLogin\_Wireframe

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" in the top right corner.

Figure 11 : SearchPage\_Zonning

The diagram shows a window titled "SellBuyAuto". The top navigation bar includes "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". It features three dropdown menus: "Marque", "Modèle", and "Motorisation". Below these are three filter boxes: "Année" with a range from 1900 to 2024, "Kilométrage" with a range from 0 to 400000+, and "Prix" with a range from 0 to 1000000+. A "Rechercher" button is located at the bottom right of the search area.

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.

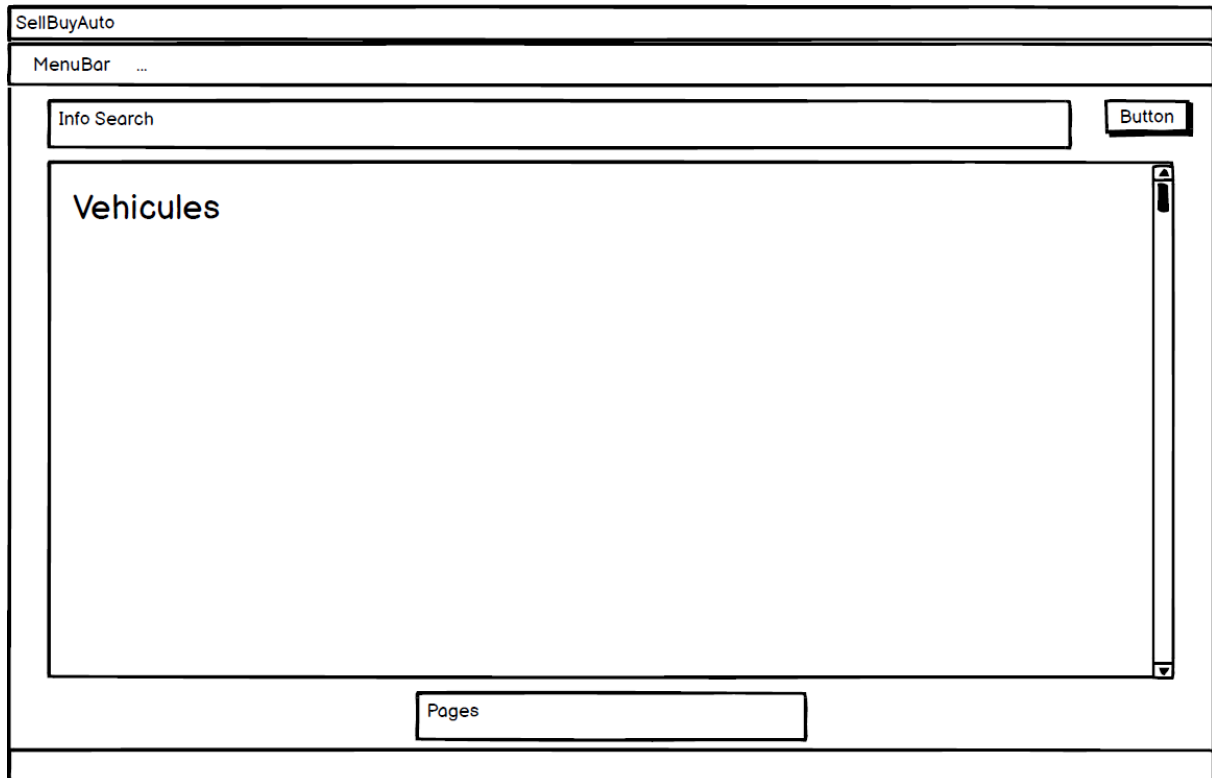


Figure 13 : VehiculesSearchPage\_Zonning

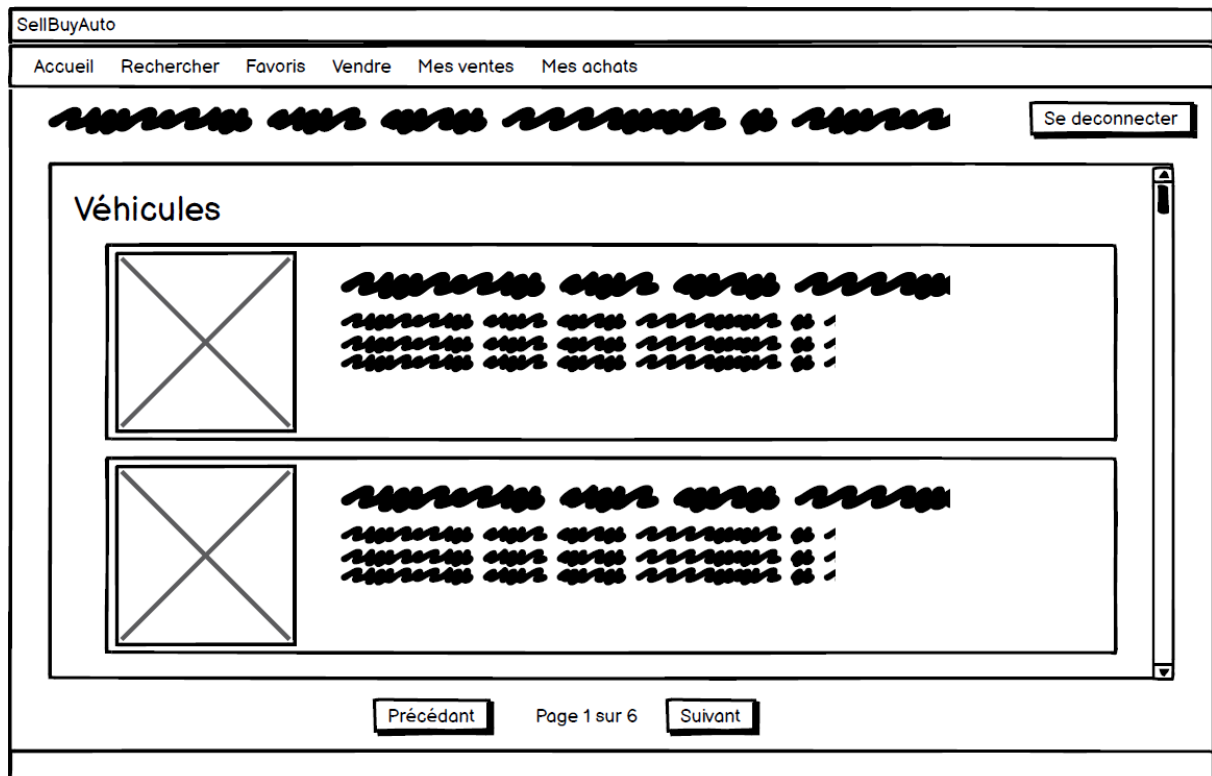


Figure 14 : VehiculesSearchPage\_Wireframe

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

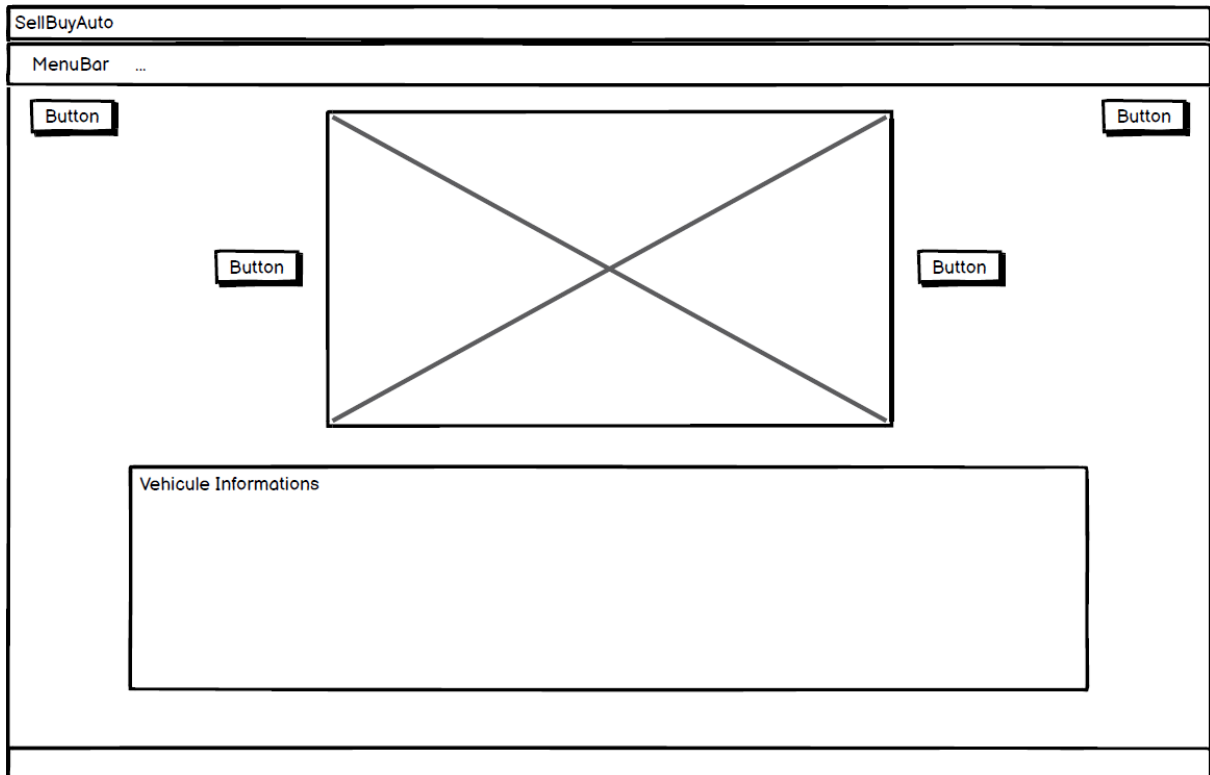


Figure 15 : VehiculeDetailPage\_Zonning

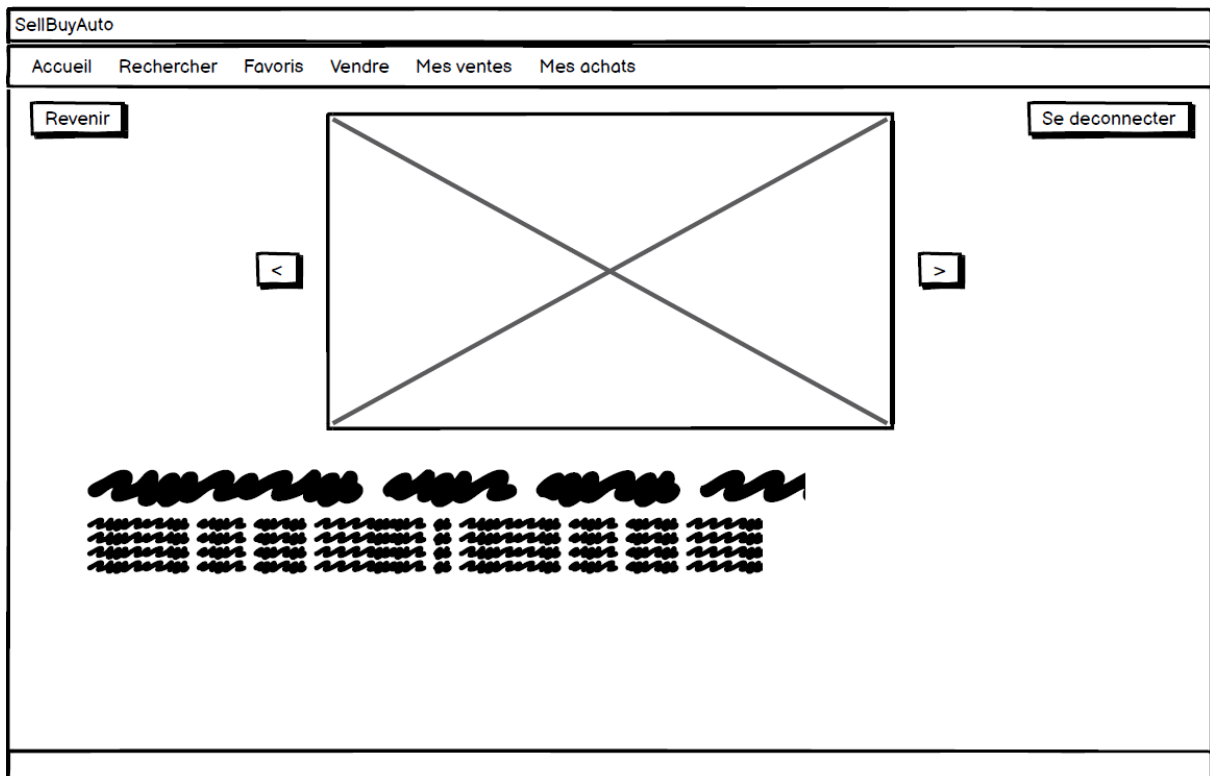


Figure 16 : VehiculeDetailPage\_Wireframe

Cette page correspond à la page d'affichage du détail d'une annonce. Elle permettra à l'utilisateur de contacter le vendeur à condition qu'il soit connecté.

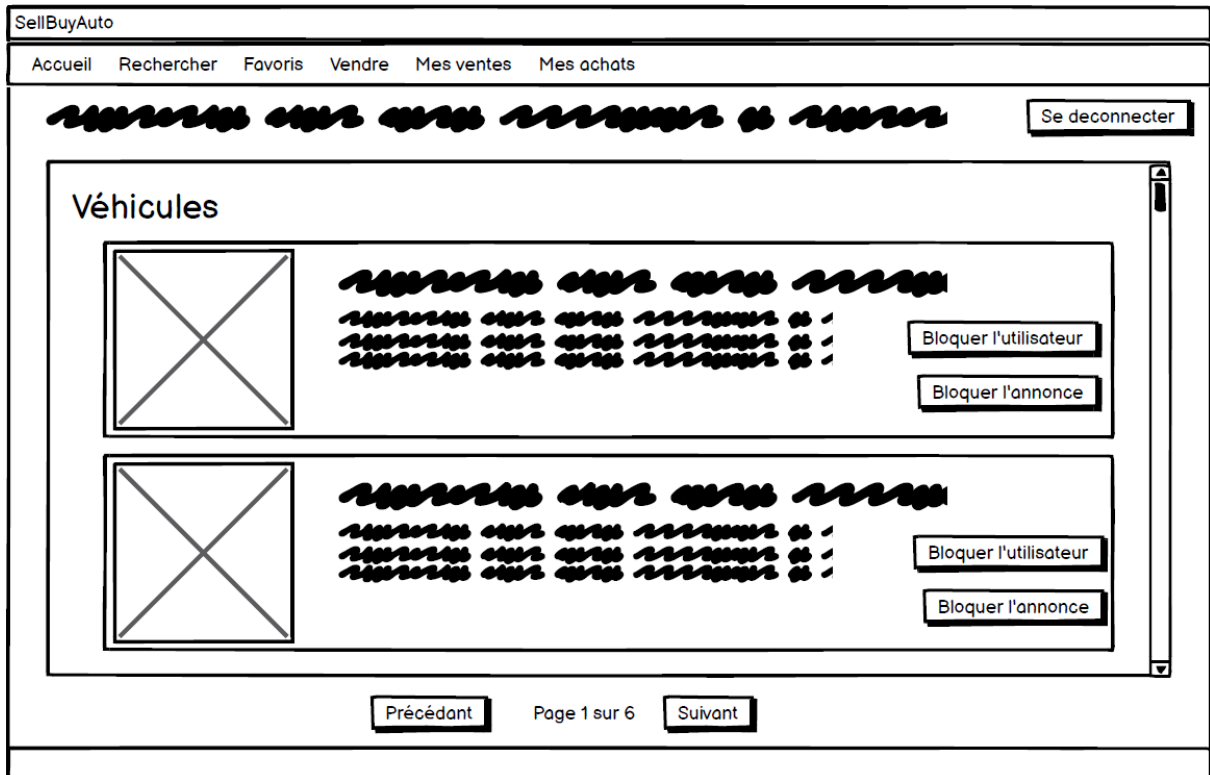


Figure 17 : VehiculesSearchPageInAdmin\_Wireframe

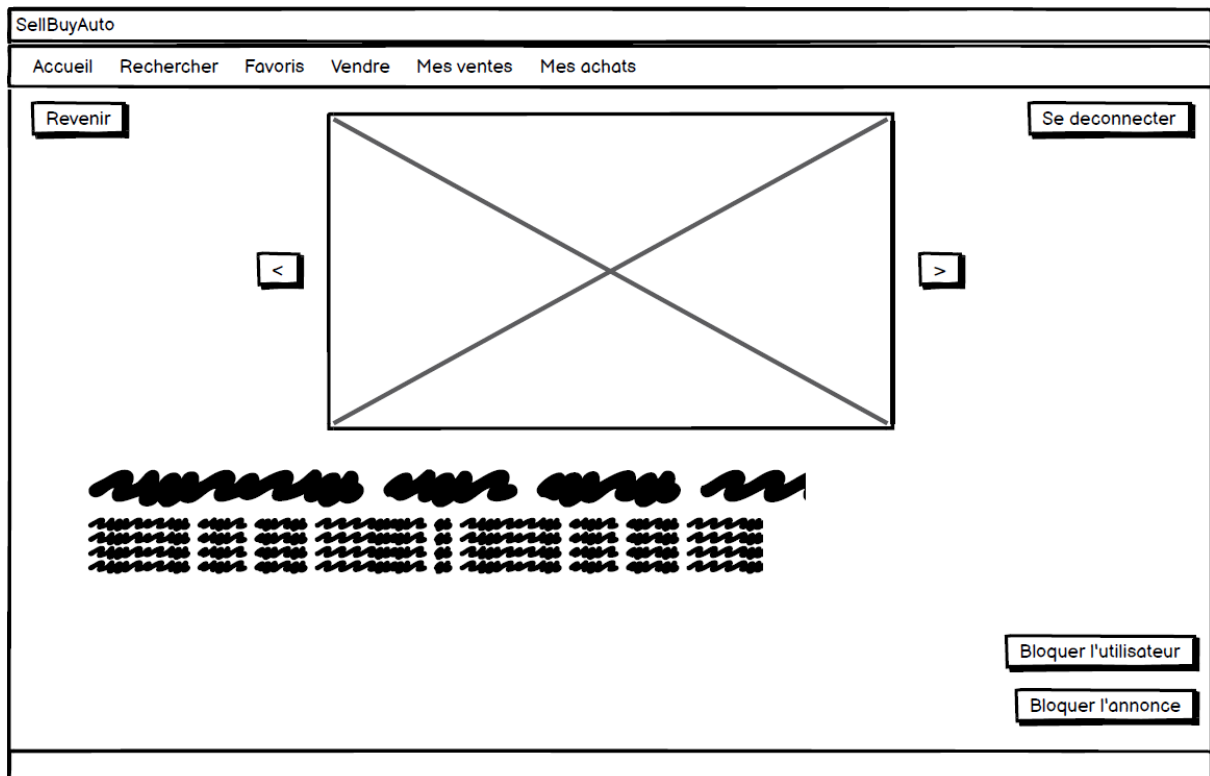


Figure 18 : VehiculeDetailPageInAdmin\_Wireframe

Ces deux pages sont les mêmes que les pages précédentes lorsque l'utilisateur qui est connecté est administrateur.

The diagram shows a web page layout for 'SellBuyAuto'. It includes a top menu bar, a large central form area, and a button in the upper right corner.

Figure 19 : AddModifyPage\_Zonning

This wireframe provides a detailed view of the 'AddModifyPage'. It features a navigation menu, a user logout button, and a comprehensive form for adding or modifying a car listing, including fields for brand, year, price, model, mileage, engine type, description, and image upload.

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.

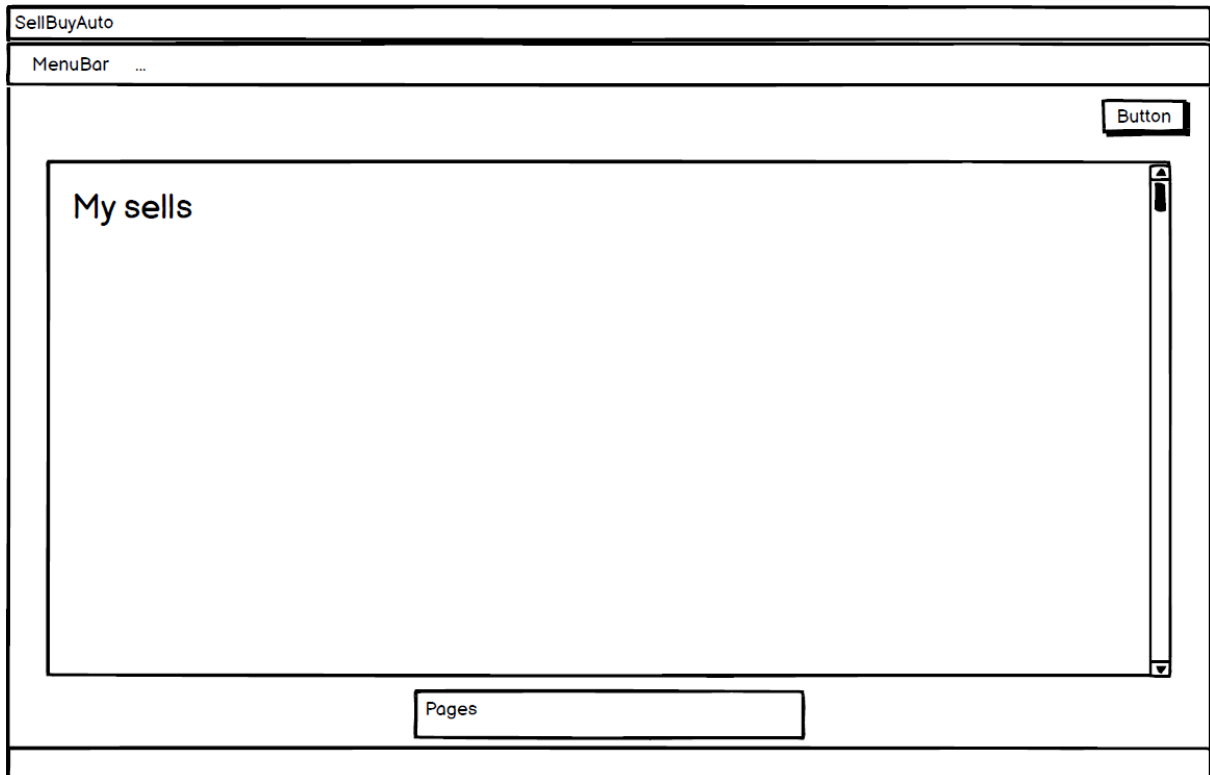


Figure 21 : MySellsPage\_Zonning

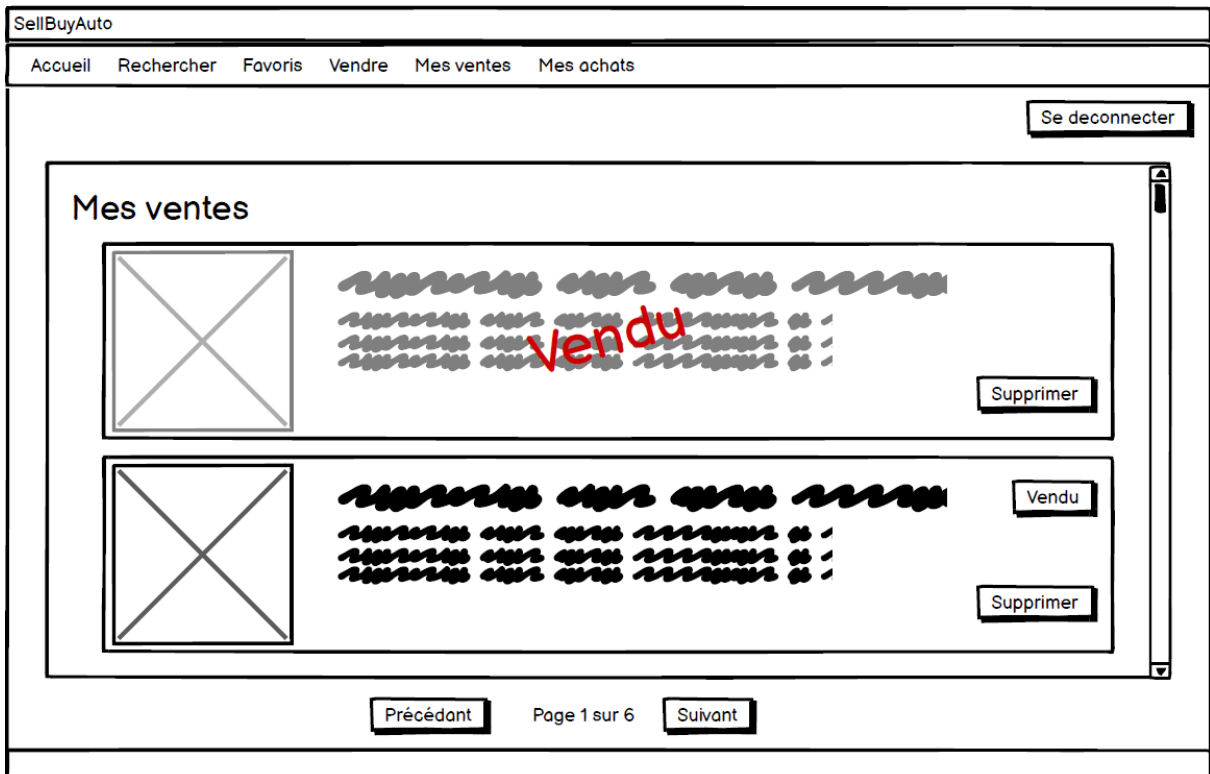


Figure 22 : MySellsPage\_Wireframe

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



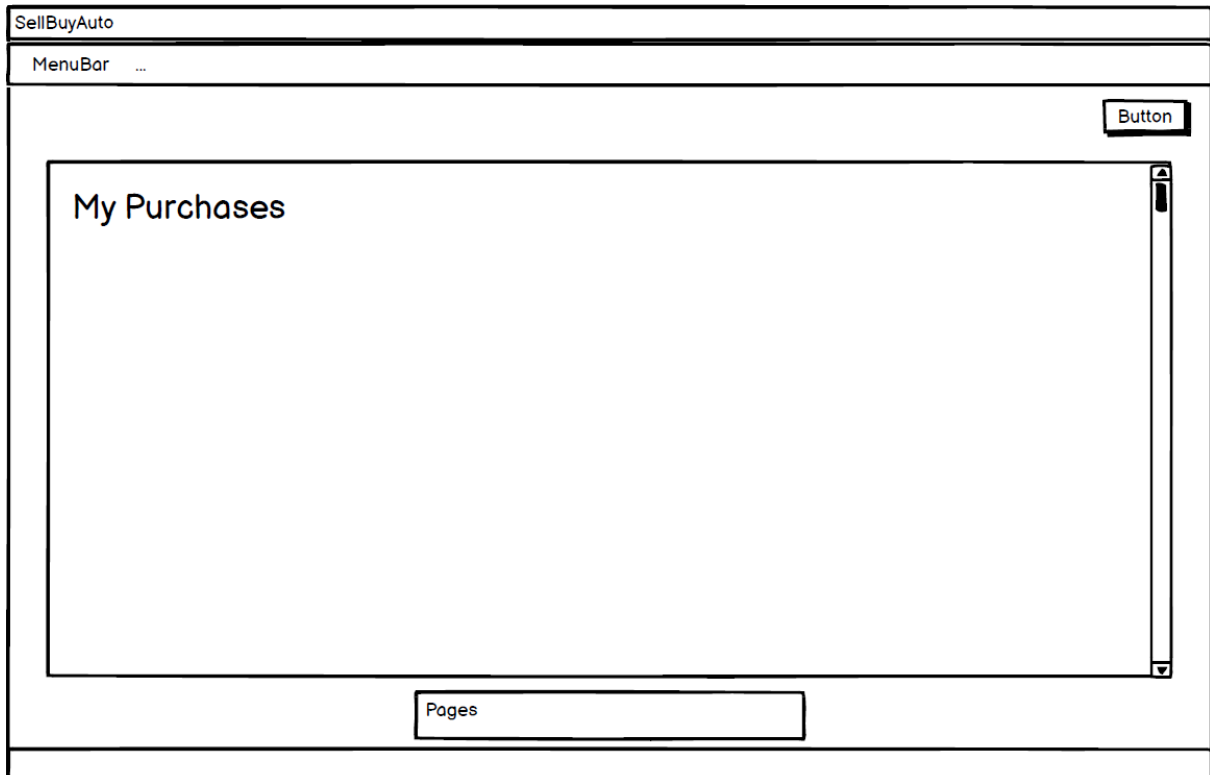


Figure 23 : MyPurchasesPage\_Zonning

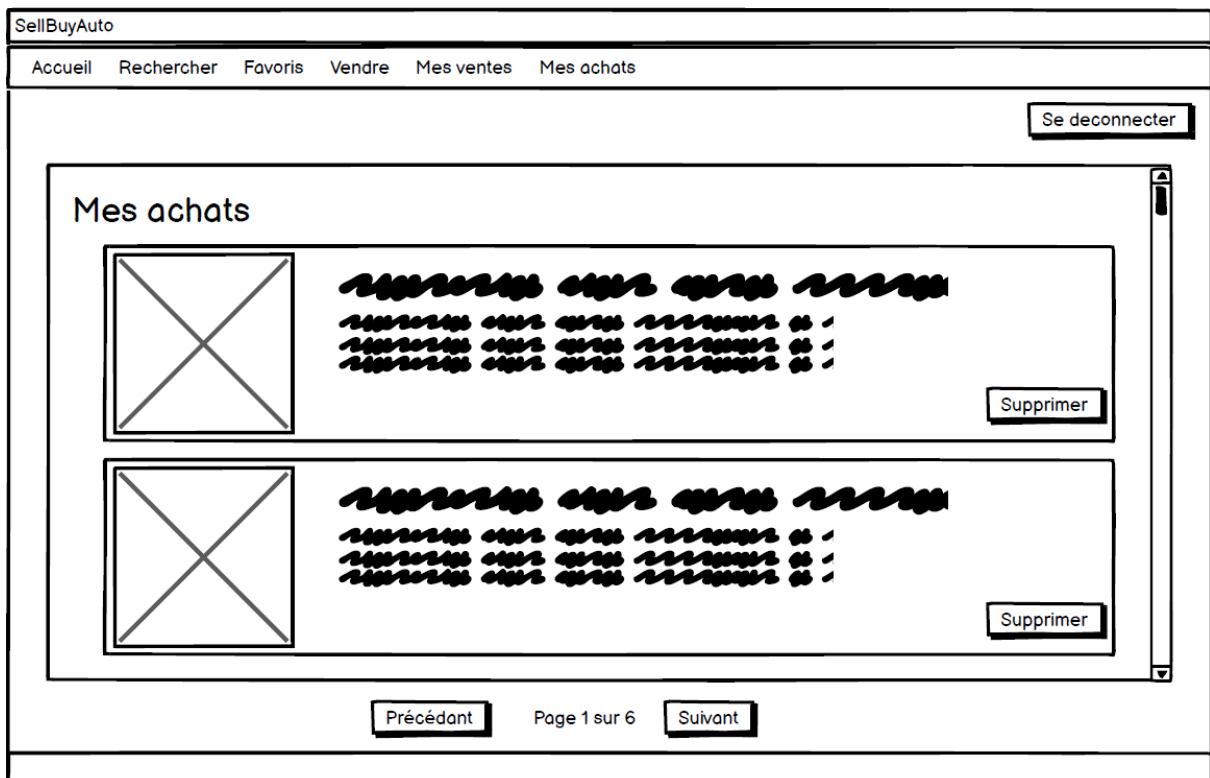


Figure 24 : MyPurchasesPage\_Wireframe

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

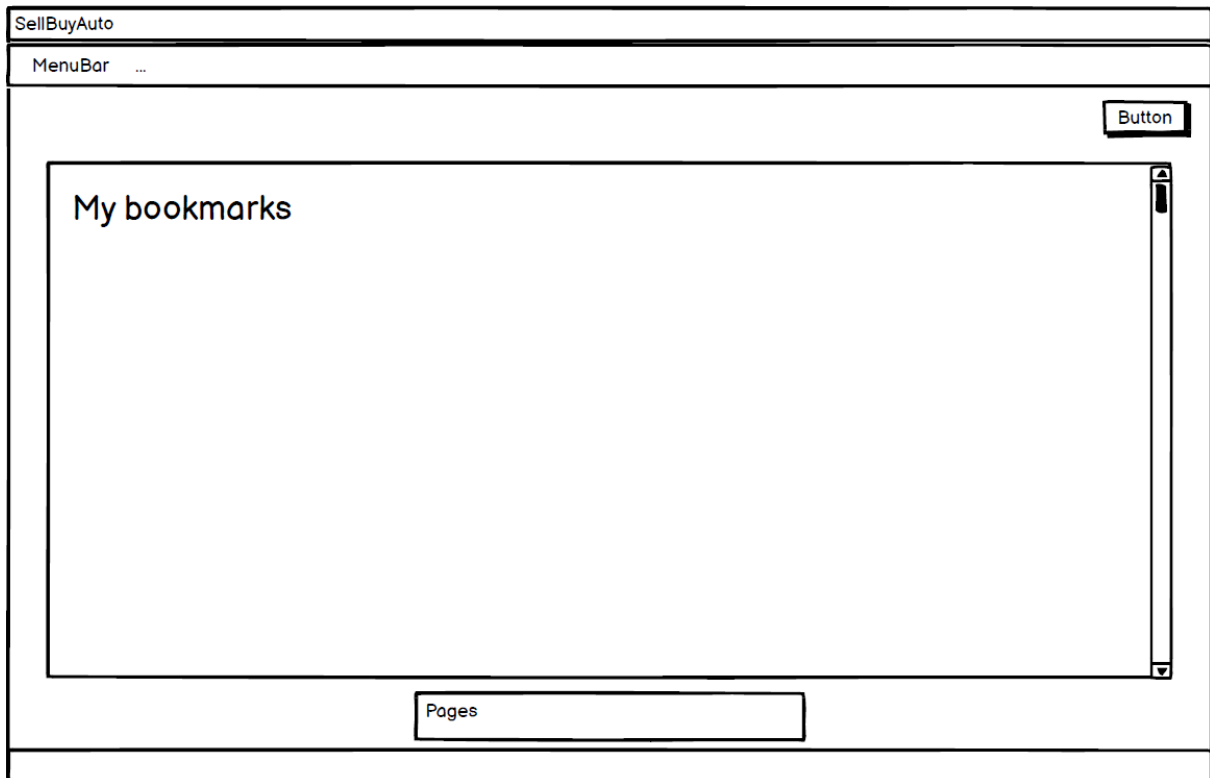


Figure 25 : MyBookmarksPage\_Zonning

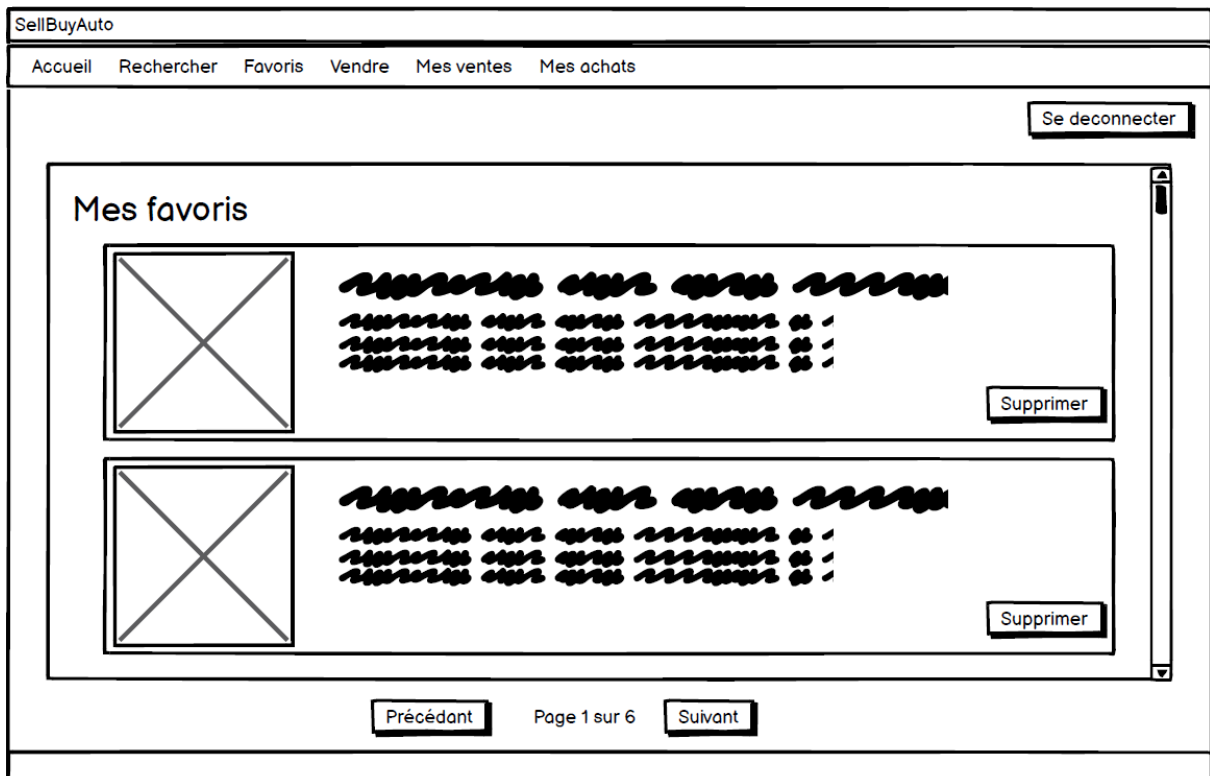


Figure 26 : MyBookmarksPage\_Wireframe

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
<b>S'enregistrer</b>	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
<b>Se connecter</b>	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
<b>Ajouter une annonce</b>	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
<b>Modifier une annonce</b>	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
<b>Supprimer une annonce</b>	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
<b>Mettre une annonce en favoris</b>	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 est affichée
<b>Consulter les annonces favorites</b>	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
<b>Consulter ses ventes</b>	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
<b>Consulter ses achats</b>	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
<b>Bloquer un utilisateur</b>	Sélectionner l'utilisateur à bloquer	L'utilisateur est un administrateur	L'utilisateur est mis comme bloquer dans la base de données et un message de confirmation s'affiche
<b>Bloquer une annonce</b>	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 testées de manière individuelle pour 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 pour 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, ainsi que les différentes maquettes de l'application, les diagrammes qui me permettront de débiter le développement et j'aimerais également que ma base de données soit créée.



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.

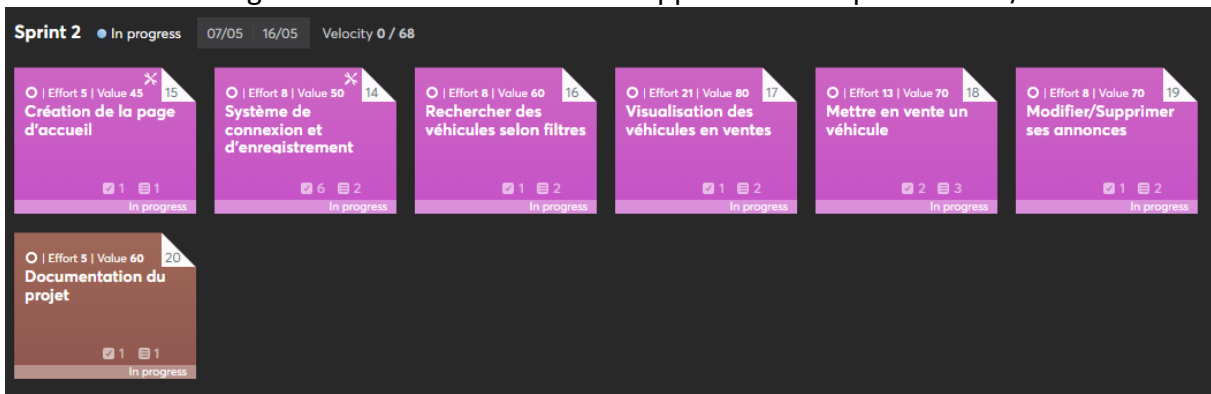


Figure 28 : Stories du sprint 2

## 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 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 disparaîtra de son compte. Cependant, si la voiture avait déjà été vendue, elle resterait accessible à l'acheteur dans la section "Mes achats". Dans le cas où un utilisateur aurait ajouté la voiture à ses favoris, une confirmation sera demandée au vendeur avant la suppression de l'annonce. Si le vendeur confirme la suppression, l'annonce sera retirée des favoris de l'utilisateur.

## 3.3 MLD

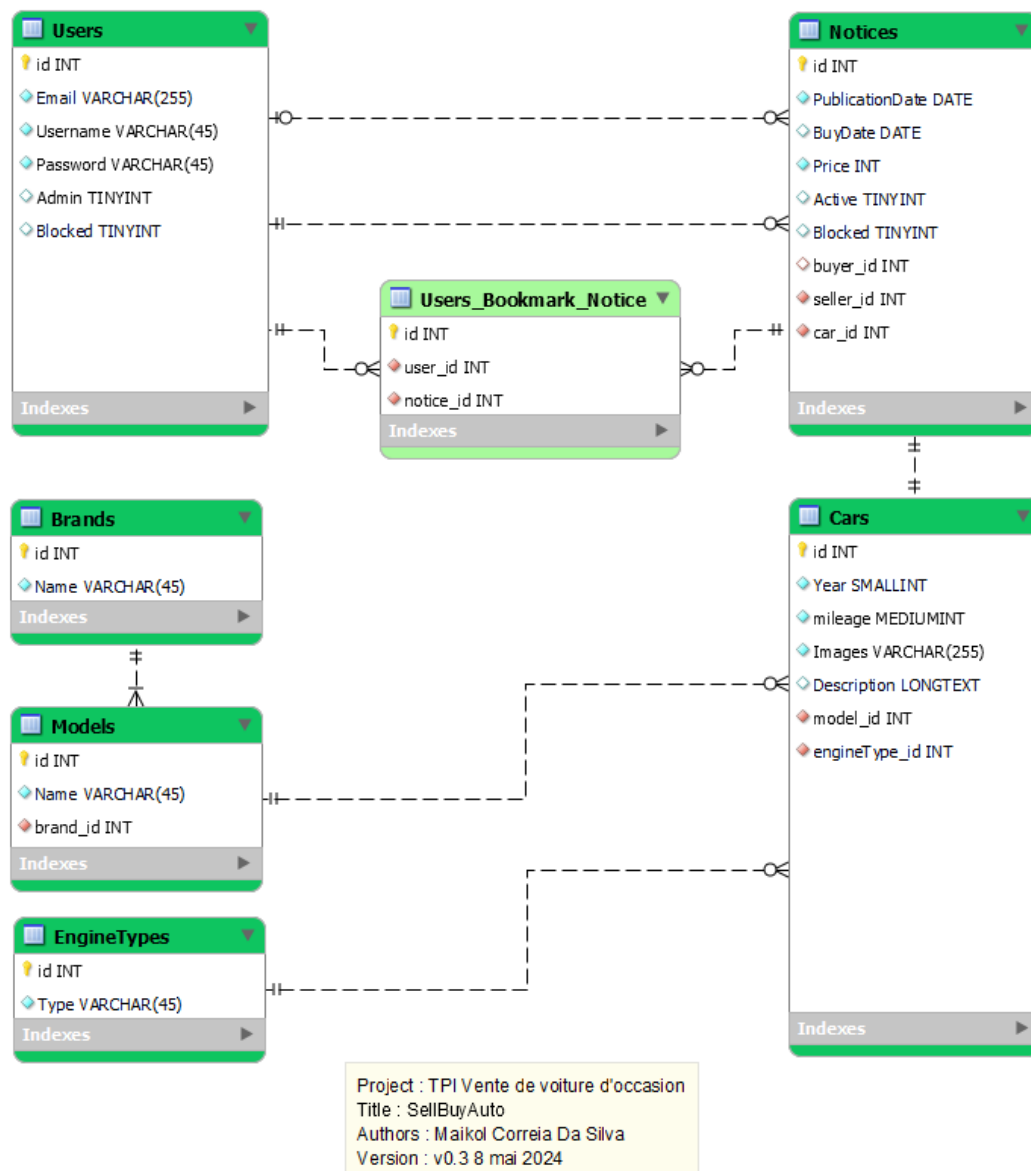


Figure 31 : MLD

Ce modèle logique de données (MLD) a été élaboré à partir du modèle conceptuel de données (MCD) présenté précédemment. 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

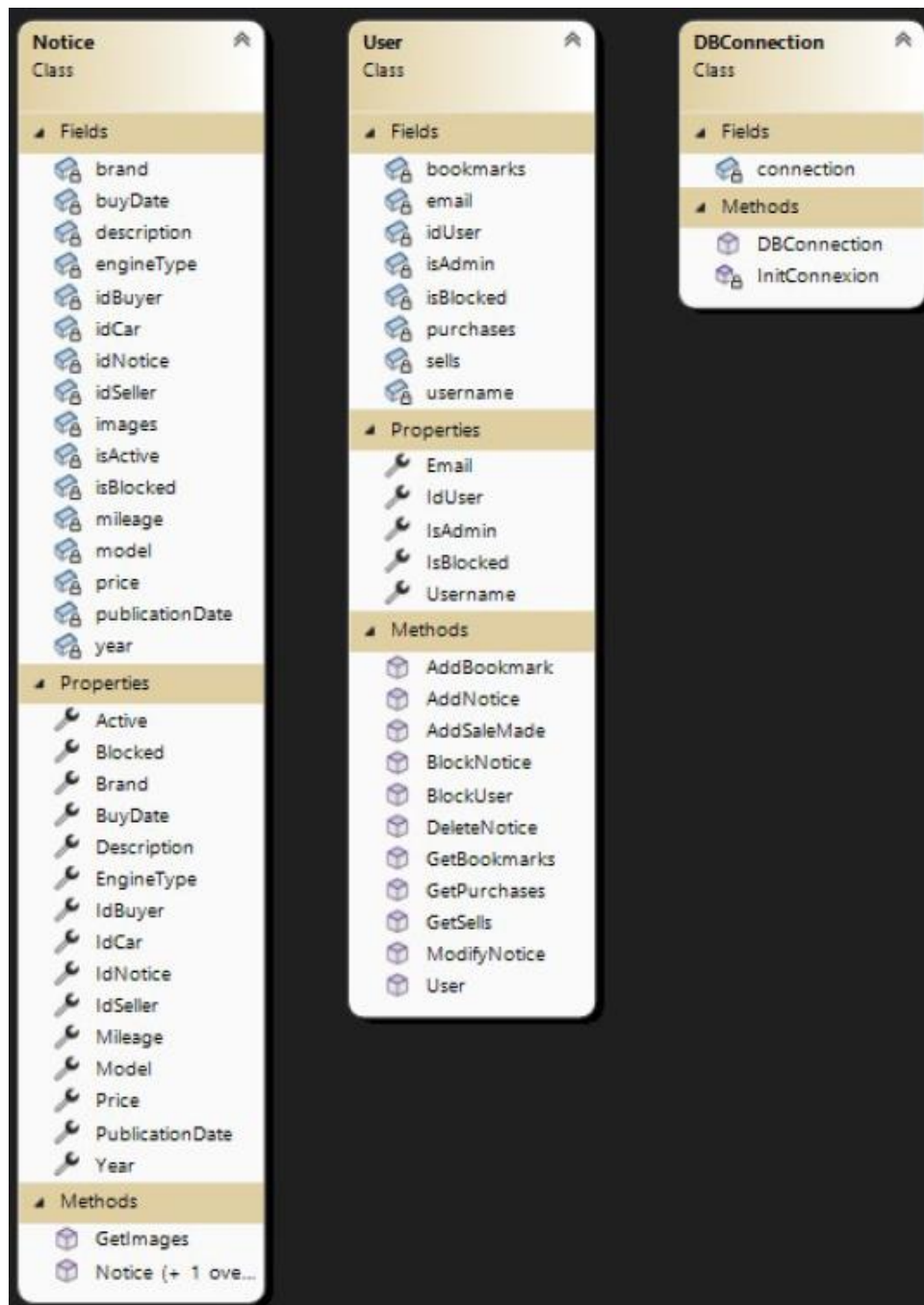


Figure 32 : Diagramme de classe

Ce projet comprend trois principales classes. 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

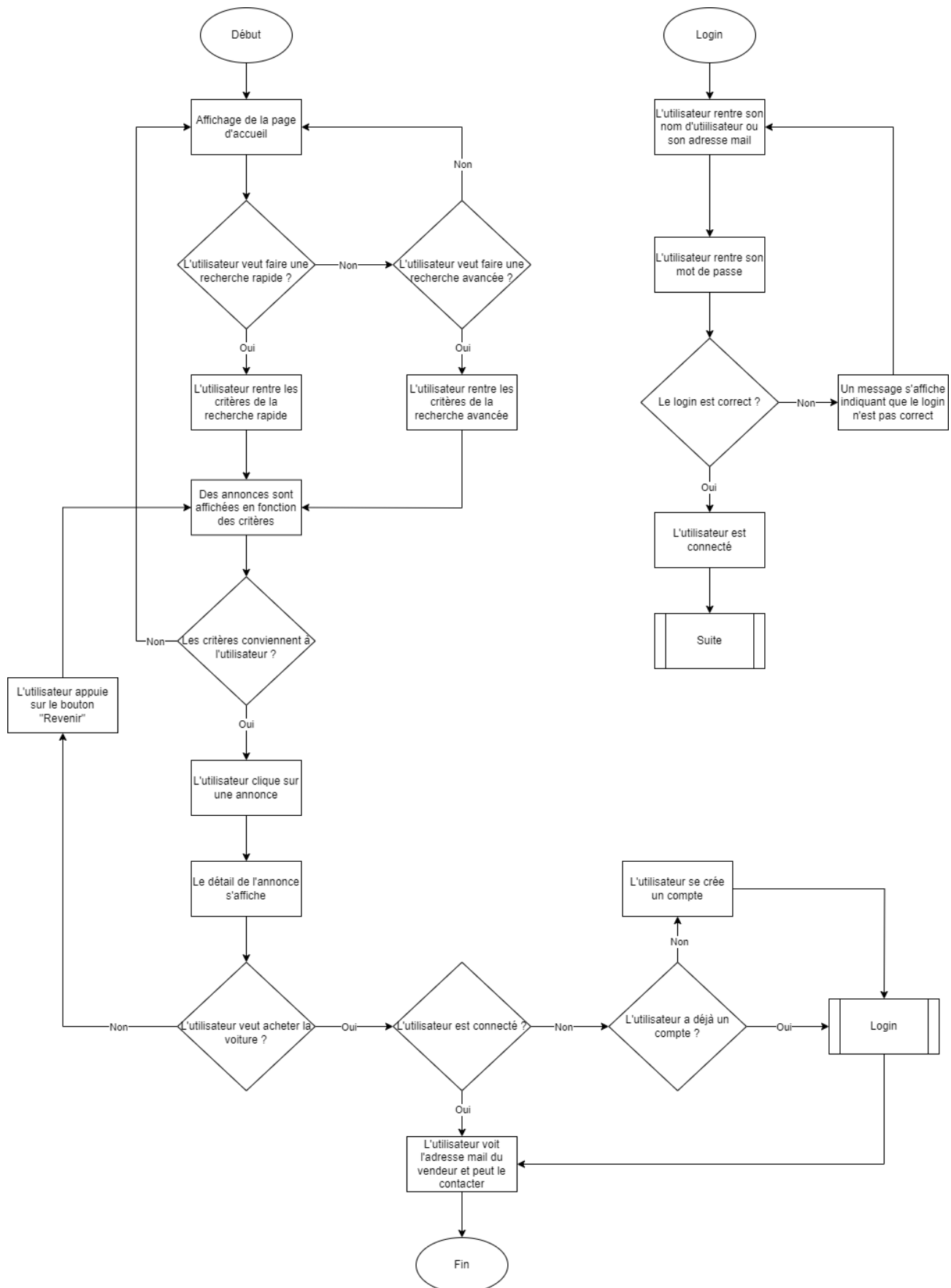


Figure 33 : Diagramme de flux

## 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. Cette documentation est un guide essentiel pour comprendre le contexte et les objectifs du projet.

**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 de la fonctionnalité de recherche d'un véhicule ainsi que fonctionnalité de connexion. Puis j'ai réalisé un diagramme de classe 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 aspect visuel.

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.

**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 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.

**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.

**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.

**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.

**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 si nécessaire.

**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 de cette page, 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. Cette documentation est importante pour expliquer mon travail, faciliter la compréhension du projet, et servir de référence pour d'éventuelles améliorations ou maintenances futures.

En résumé, ce sprint a été riche en développements, avec la création de plusieurs fonctionnalités clés qui ont considérablement amélioré l'expérience utilisateur et la gestion des véhicules.

#### 4.1.4 Sprint 4

### 4.2 Description des tests effectués

Tests	Action	Conditions	Réaction attendue	Réaction obtenue	Validé
<b>S'enregistrer</b>	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
<b>Se connecter</b>	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
<b>Ajouter une annonce</b>	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
<b>Modifier une annonce</b>	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

## Application C# de type plateforme de vente de voitures

		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
<b>Supprimer une annonce</b>	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
<b>Mettre une annonce en favoris</b>	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 est affichée	Oui
<b>Consulter les annonces favorites</b>	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
<b>Consulter ses ventes</b>	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
<b>Consulter ses achats</b>	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



## Application C# de type plateforme de vente de voitures

		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
<b>Bloquer un utilisateur</b>	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		
<b>Bloquer une annonce</b>	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		

Tableau 2: Test effectués

## 4.3 Test de l'Application par un Utilisateur Réel

## 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 les 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.

## 5 Conclusion

## 6 Annexes

### 6.1 Résumé du rapport du Pré-TPI

### 6.2 Glossaire

### 6.3 Sources – Bibliographie

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

Run méthode en arrière-plan : <https://stackoverflow.com/questions/20304258/run-async-method-on-a-background-thread>

Sup variable de la mémoire : <https://security.stackexchange.com/questions/212917/how-to-delete-a-c-variable-value-permanently-from-memory>

Ajouter des control dans un flow layout panel : <https://www.youtube.com/watch?v=VeapnO7b2gl>

Limiter les filtres dans openFileDialog : <https://stackoverflow.com/questions/2069048/setting-the-filter-to-an-openfiledialog-to-allow-the-typical-image-formats>

Message de confirmation avant de supprimer : <https://www.youtube.com/watch?v=rjETrJG0a0w>

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

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

Lancer l'événement click avec le code : <https://bytes.com/topic/c-sharp/472035-how-programmatically-invoke-click-event-button>

Copier du texte dans le presse-papiers : <https://stackoverflow.com/questions/3546016/how-to-copy-data-to-clipboard-in-c-sharp>

### 6.4 Journal de Travail

### 6.5 Code du script de génération de données