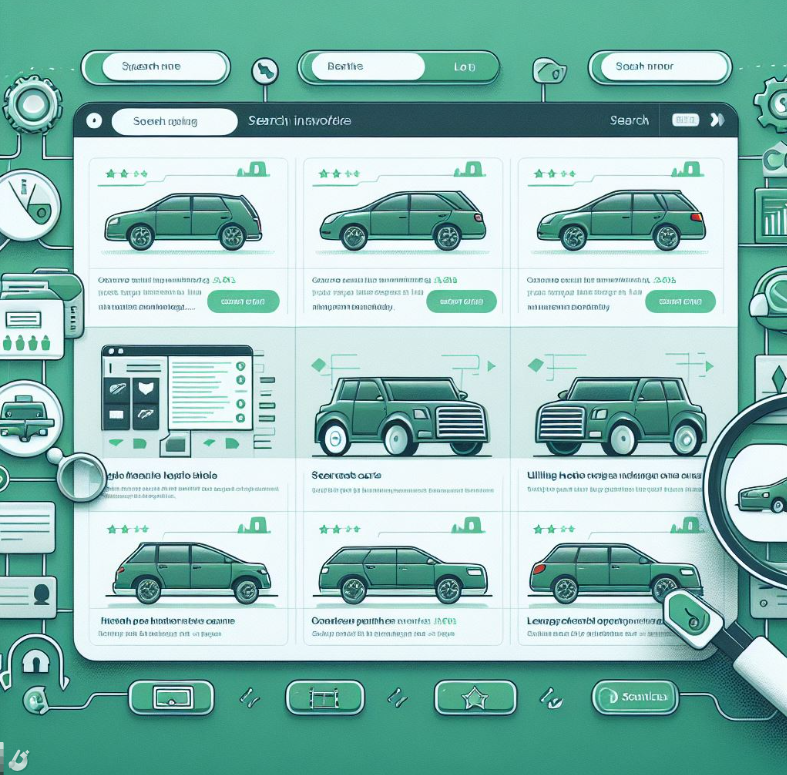
|  |
| --- |
| Application C# de type plateforme de vente de voitures |



*Maikol Correia Da Silva*

*Maikol.Correia@eduvaud.ch*



Table des matières

SI-CA2a

*06.05.2024*

[1 Introduction 3](#_Toc165900401)

[1.1 Cadre, description et motivation 3](#_Toc165900402)

[1.2 Organisation 3](#_Toc165900403)

[1.3 Objectifs 3](#_Toc165900404)

[1.4 Planification initiale 4](#_Toc165900405)

[2 Analyse 6](#_Toc165900406)

[2.1 Cahier des charges détaillé 6](#_Toc165900407)

[2.1.1 MCD 6](#_Toc165900408)

[2.1.2 Maquettes 7](#_Toc165900409)

[2.1.3 Use Cases et scénarios 19](#_Toc165900410)

[2.2 Stratégie de test 20](#_Toc165900411)

[2.3 Planification 21](#_Toc165900412)

[2.3.1 Sprint 1 21](#_Toc165900413)

[2.3.2 Sprint 2 21](#_Toc165900414)

[2.3.3 Sprint 3 21](#_Toc165900415)

[2.3.4 Sprint 4 21](#_Toc165900416)

[3 Conception 22](#_Toc165900417)

[3.1 Logiciels 22](#_Toc165900418)

[3.2 Choix techniques 22](#_Toc165900419)

[3.3 MLD 23](#_Toc165900420)

[3.4 Diagramme de classes 24](#_Toc165900421)

[3.5 Diagramme de flux 25](#_Toc165900422)

[4 Réalisation 26](#_Toc165900423)

[4.1 Dossier de réalisation 26](#_Toc165900424)

[4.1.1 Sprint 1 26](#_Toc165900425)

[4.1.2 Sprint 2 26](#_Toc165900426)

[4.1.3 Sprint 3 26](#_Toc165900427)

[4.1.4 Sprint 4 26](#_Toc165900428)

[4.2 Description des tests effectués 26](#_Toc165900429)

[4.3 Test de l'Application par un Utilisateur Réel 26](#_Toc165900430)

[4.4 Erreurs restantes 26](#_Toc165900431)

[5 Conclusion 26](#_Toc165900432)

[6 Annexes 26](#_Toc165900433)

[6.1 Résumé du rapport du Pré-TPI 26](#_Toc165900434)

[6.2 Glossaire 26](#_Toc165900435)

[*6.3* Sources – Bibliographie 26](#_Toc165900436)

[6.4 Journal de Travail 26](#_Toc165900437)

[6.5 Code du script de génération de données 26](#_Toc165900438)

# Introduction

## 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.

## 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.

## 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

## Planification initiale

Le projet a débuté le 30 avril 2024 à 8h00 et se terminera 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.

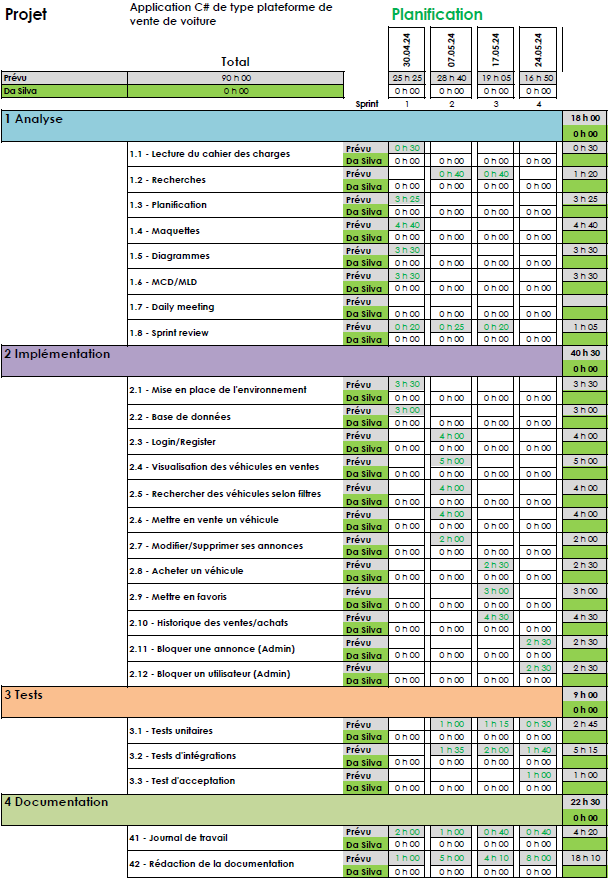


Figure 1 : Planification Initiale

# Analyse

## Cahier des charges détaillé

### MCD

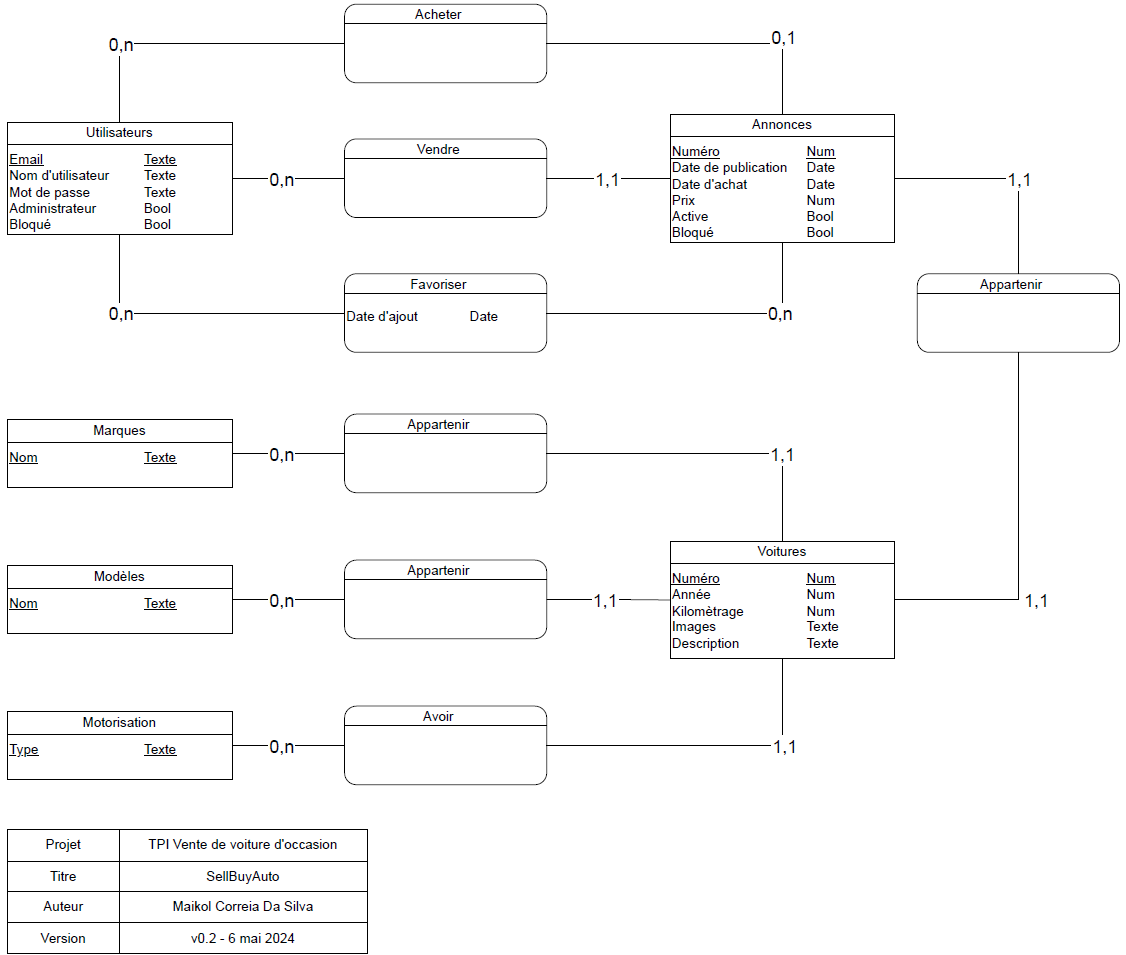


Figure : MCD

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.

### Maquettes

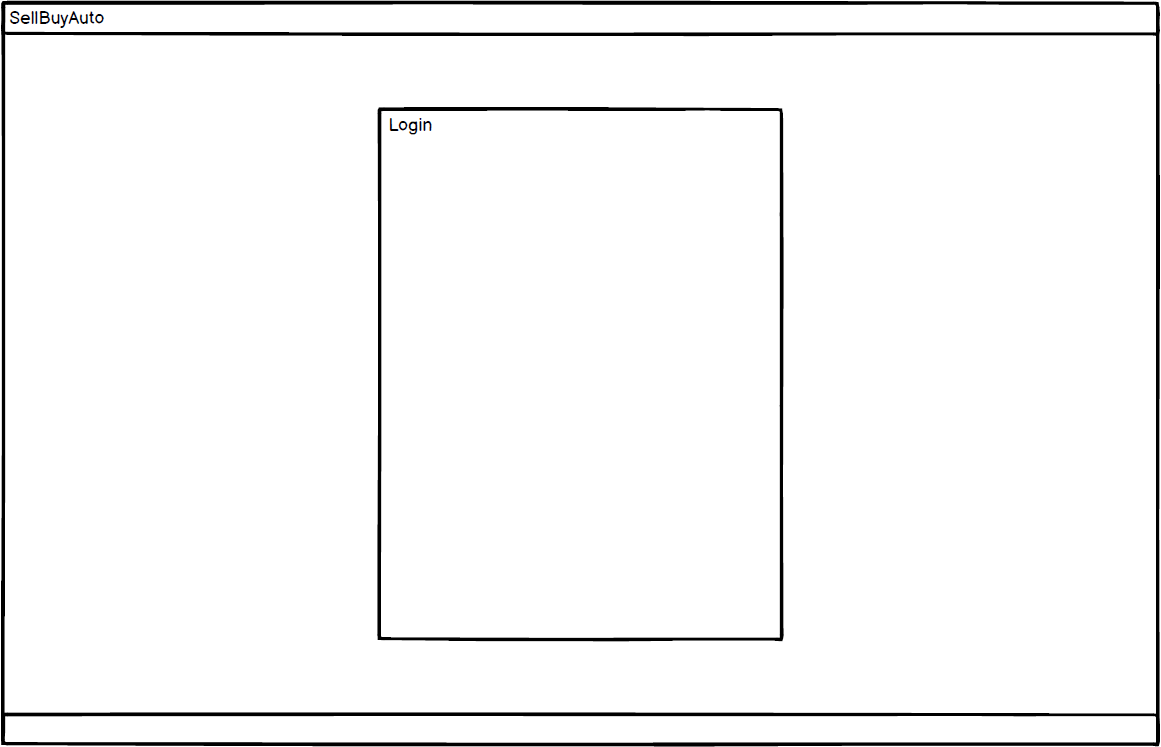


Figure : LoginPage\_Zonning

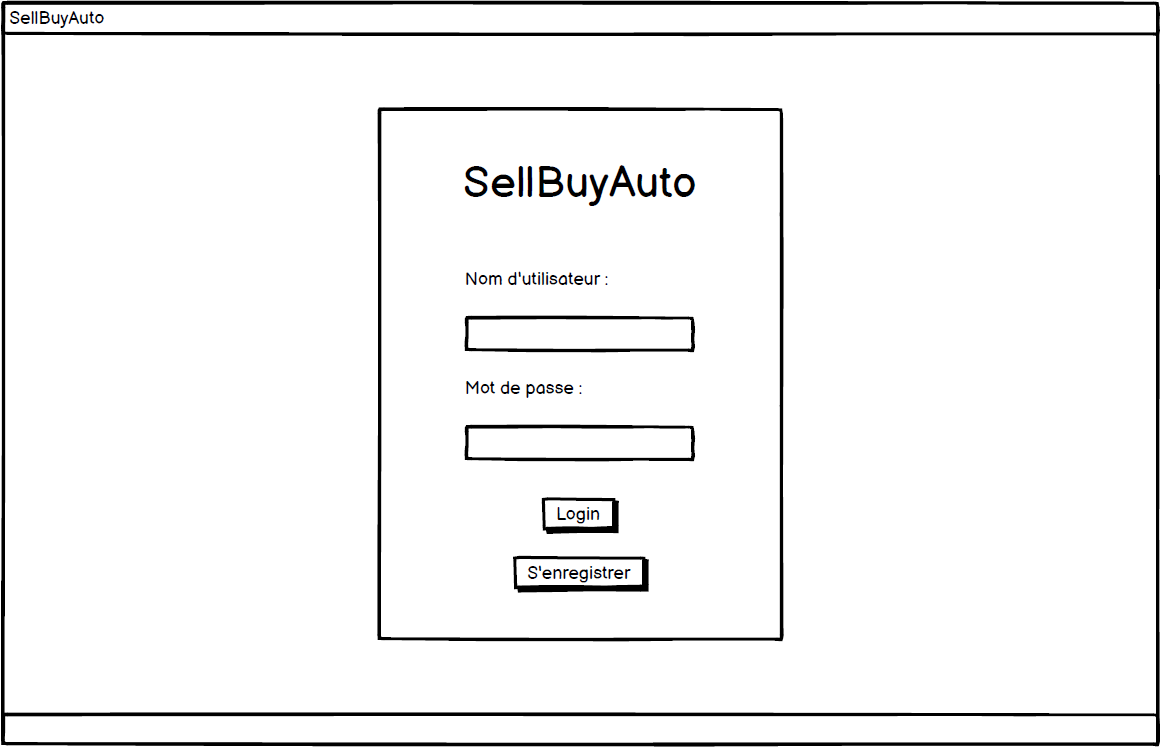


Figure : LoginPage\_Wireframe

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

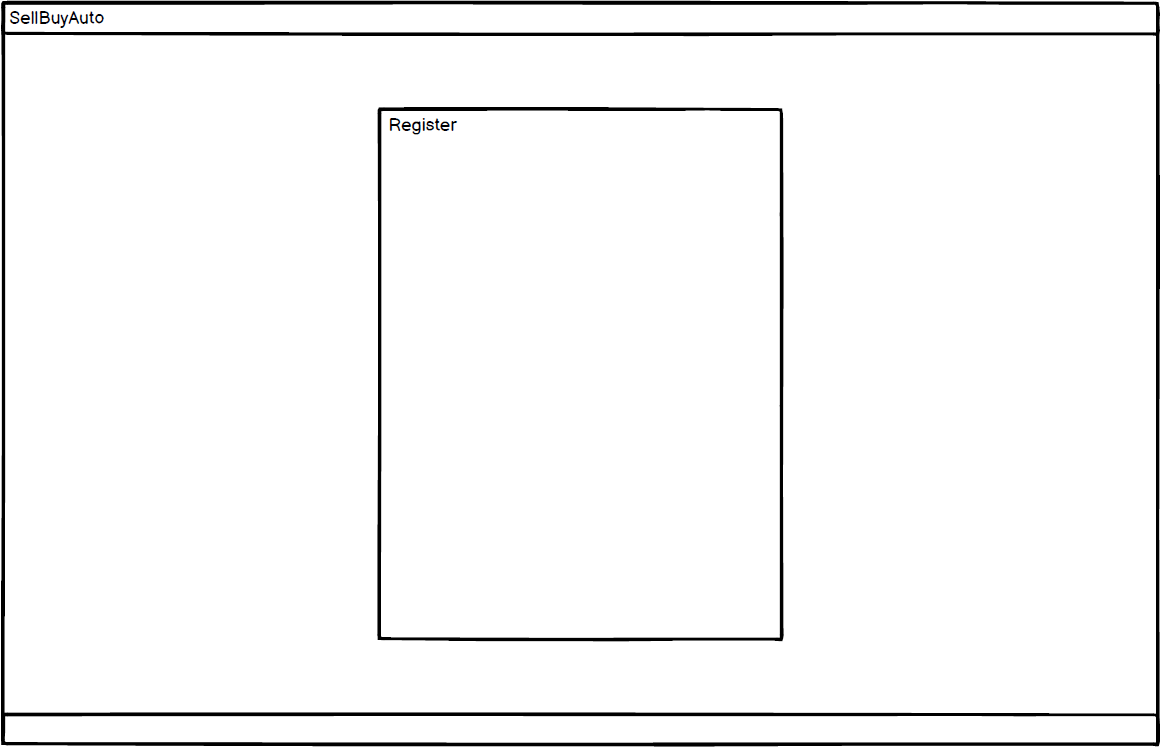


Figure : RegisterPage\_Zonning

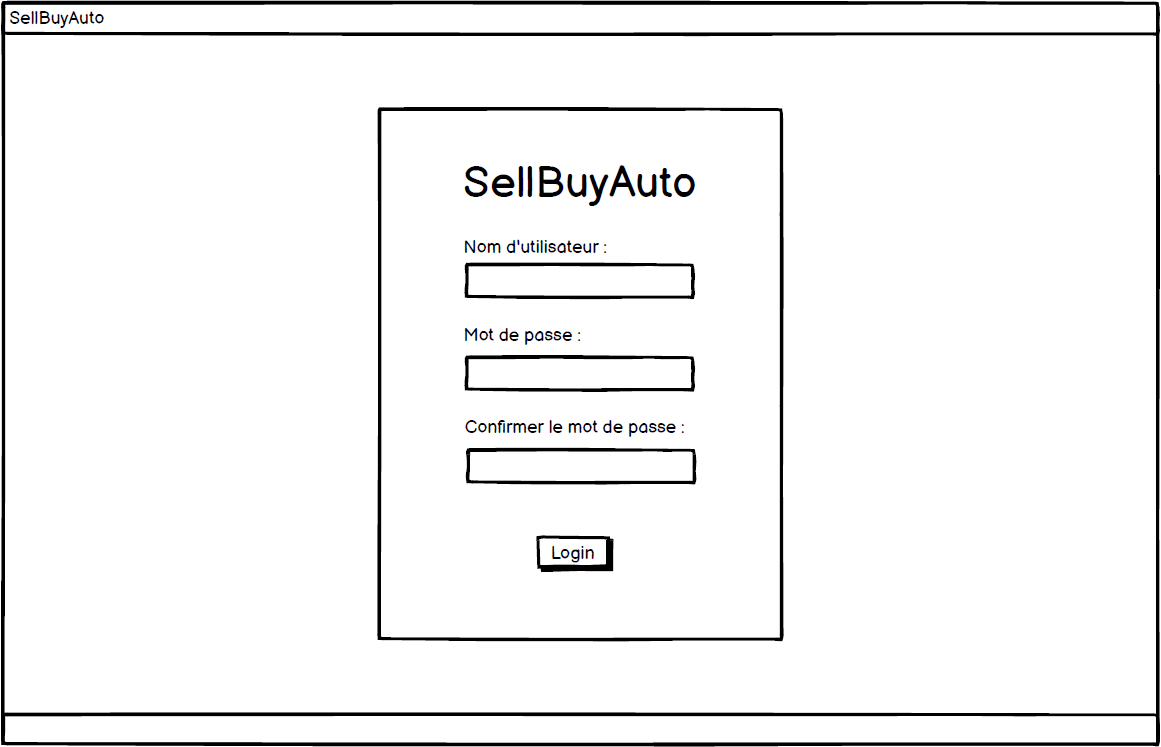


Figure : RegisterPage\_Wireframe

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

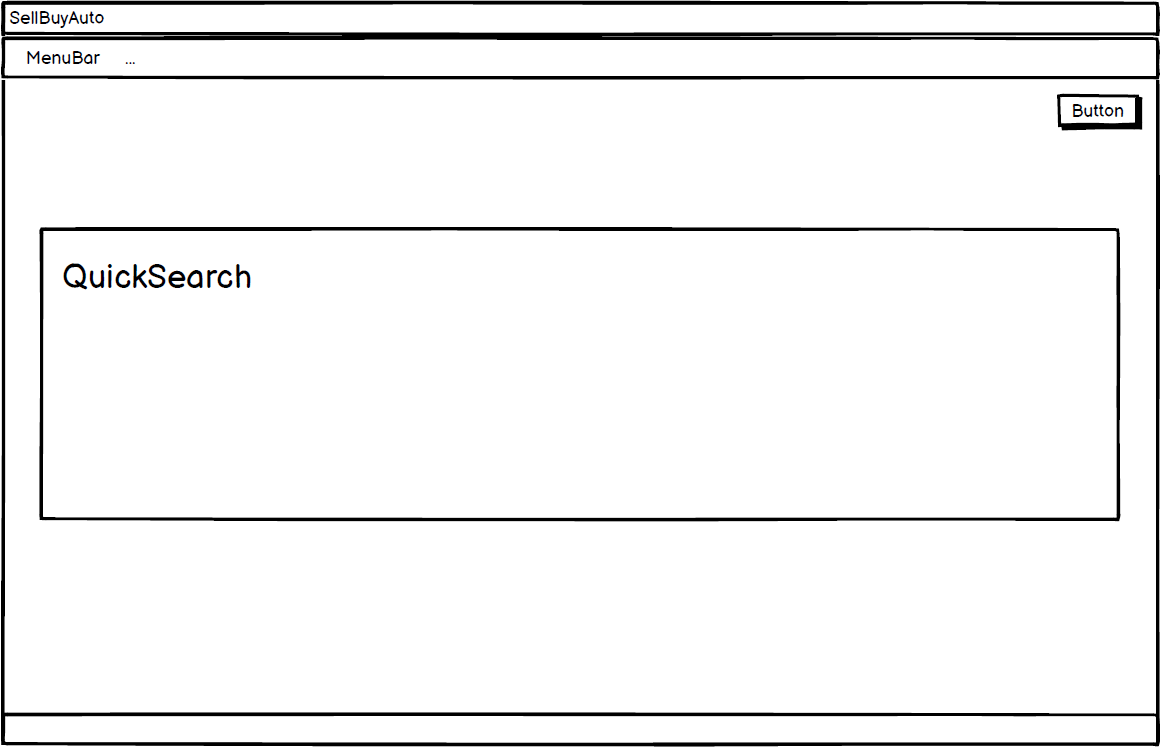


Figure : HomePageWithoutLogin\_Zonning

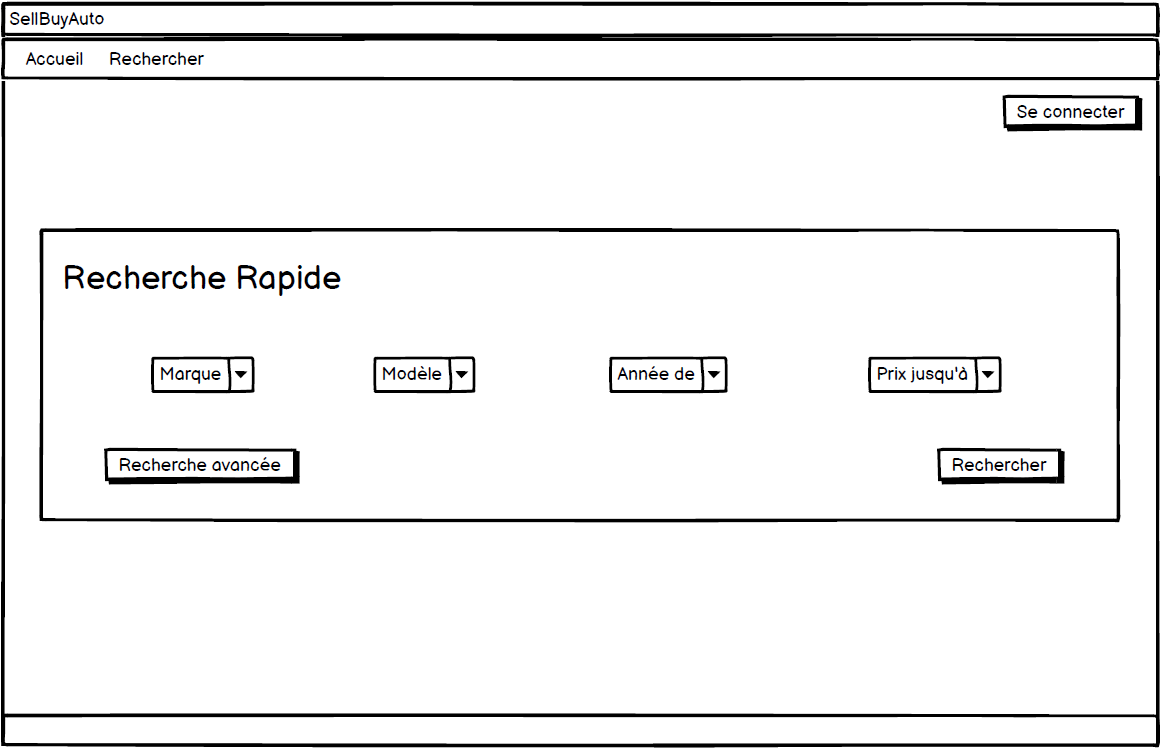


Figure : HomePageWithoutLogin\_Wireframe

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.

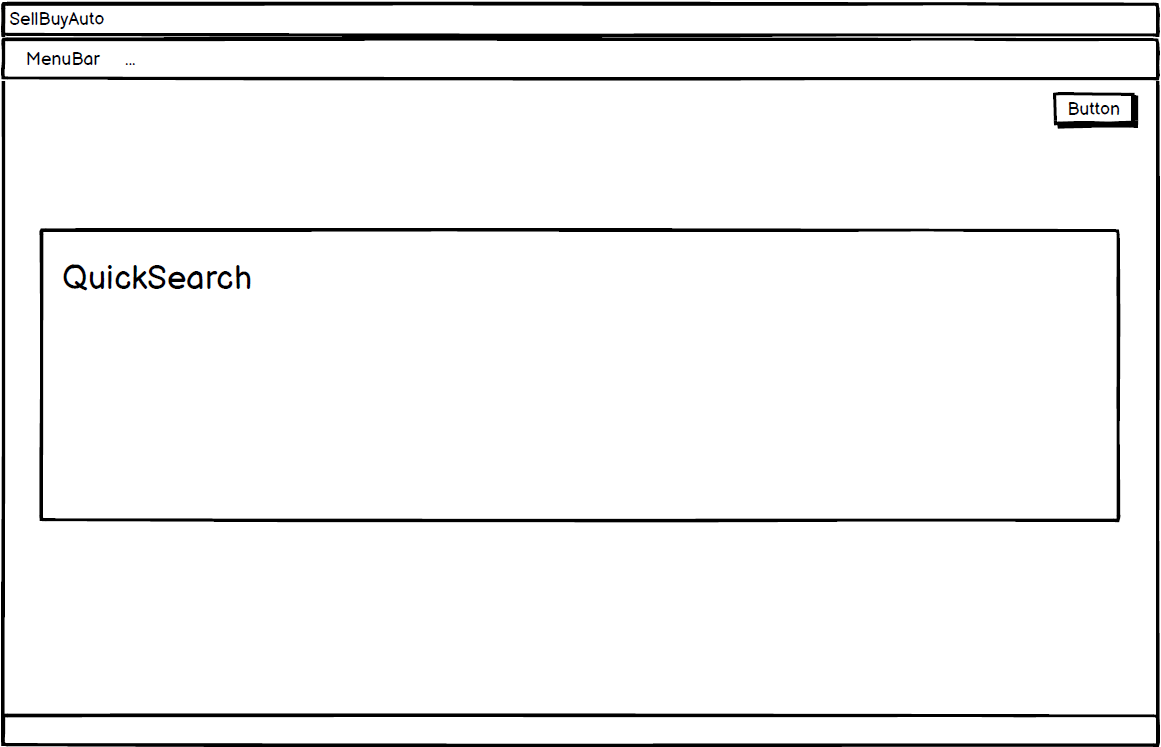


Figure : HomePageWithLogin\_Zonning

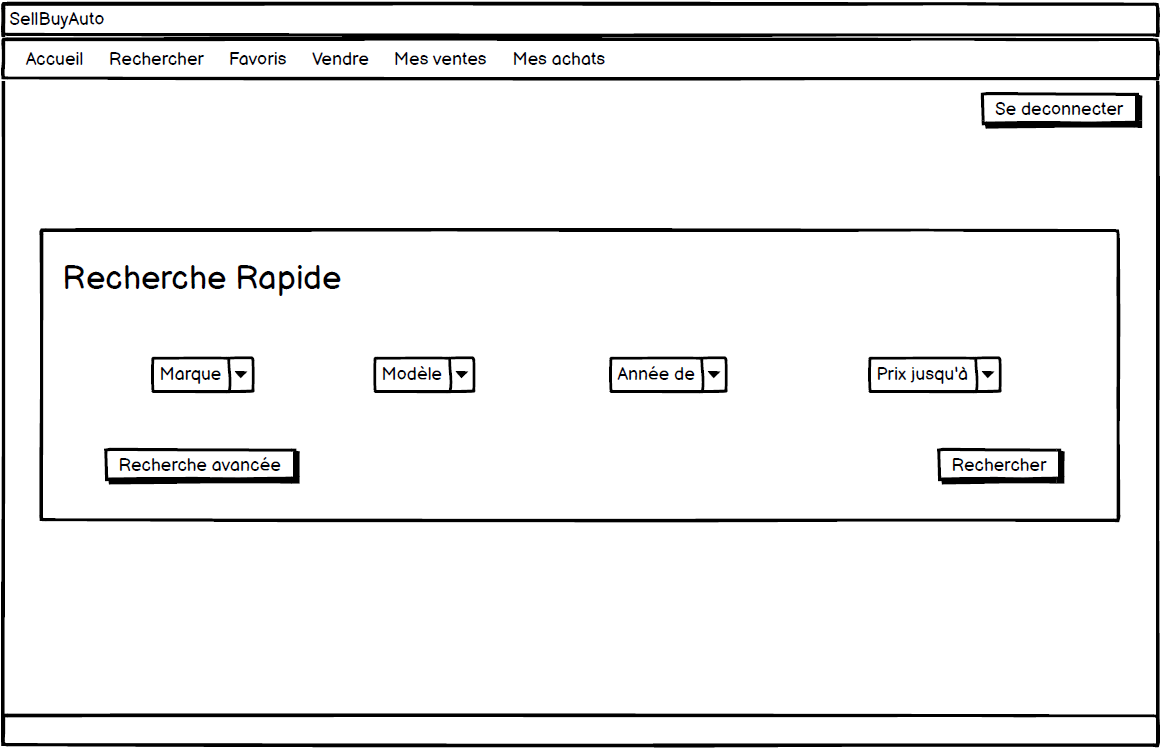


Figure : HomePageWithLogin\_Wireframe

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

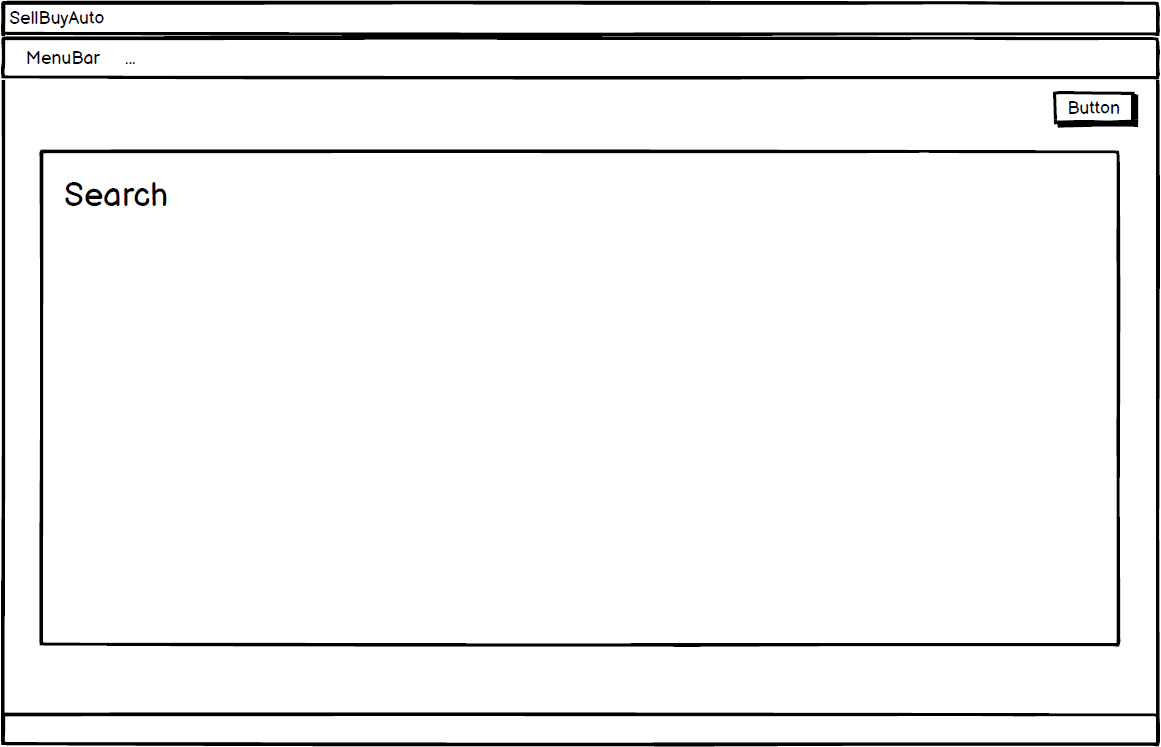


Figure : SearchPage\_Zonning

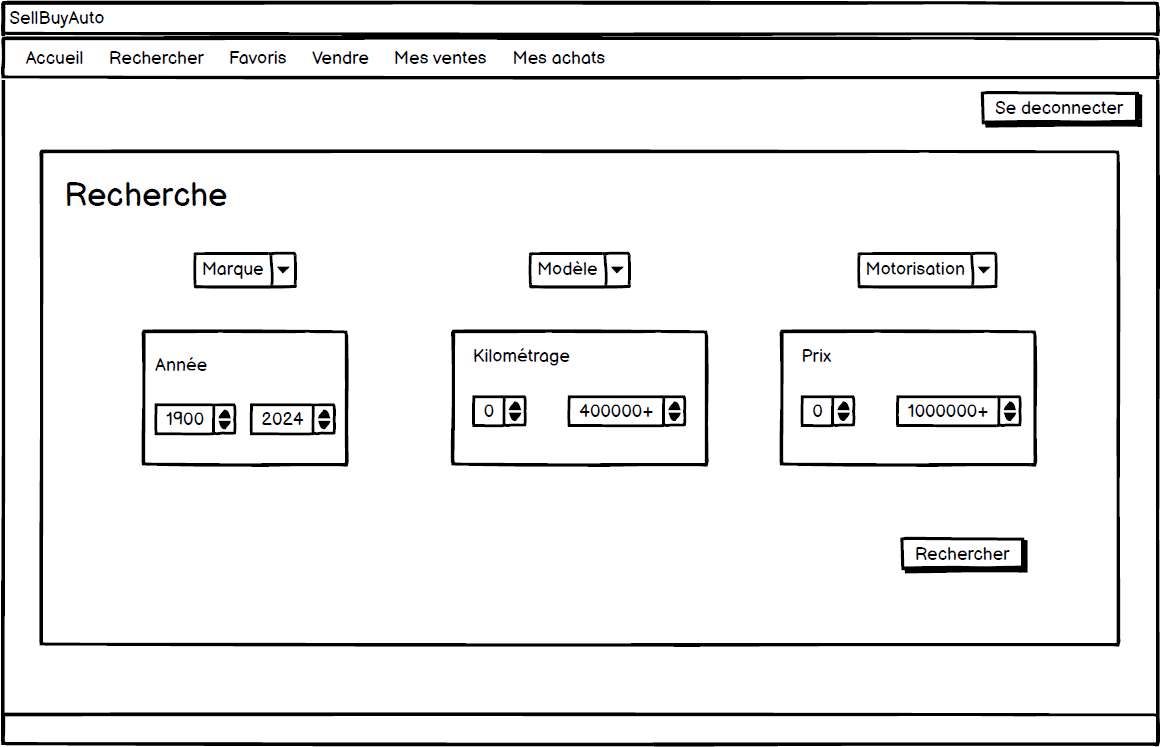


Figure : SearchPage\_Wireframe

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.

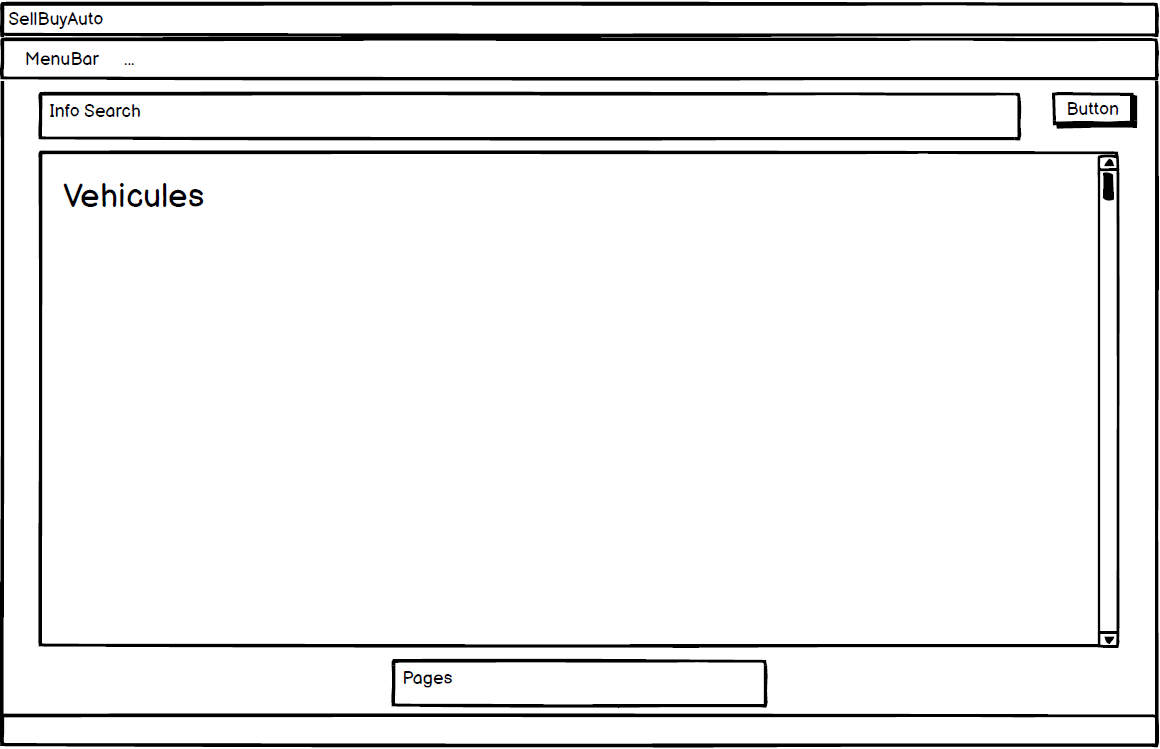


Figure : VehiculesSearchPage\_Zonning

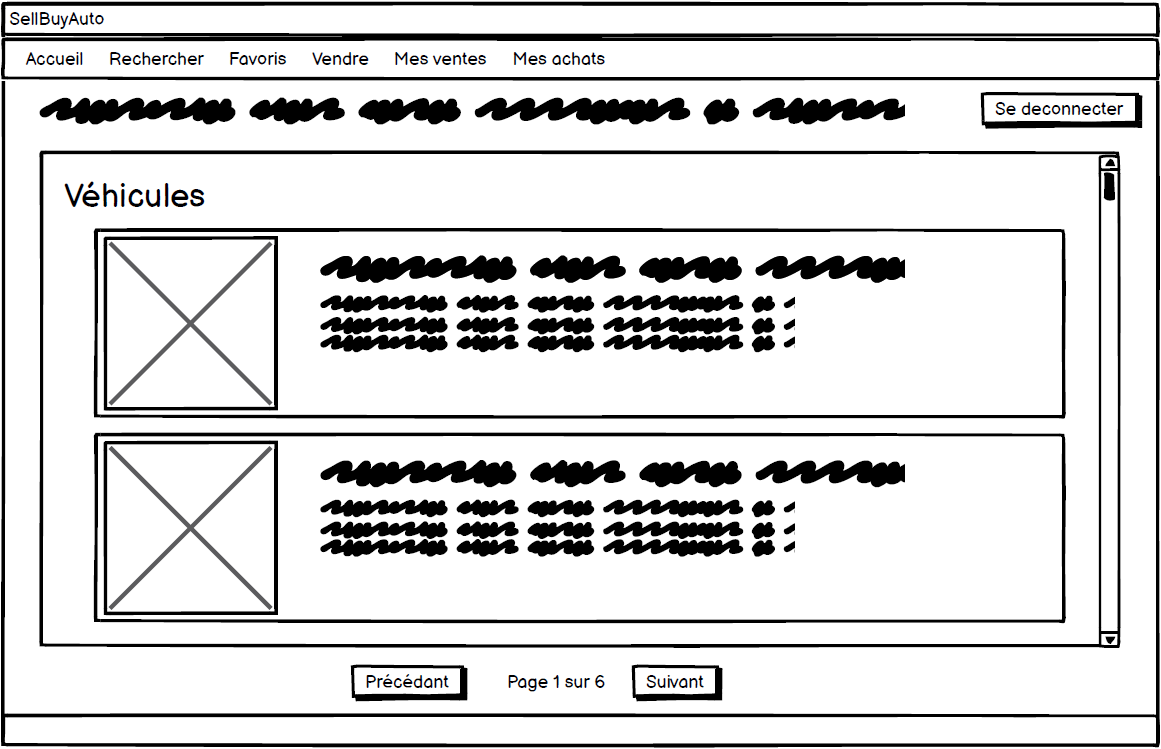


Figure : VehiculesSearchPage\_Wireframe

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

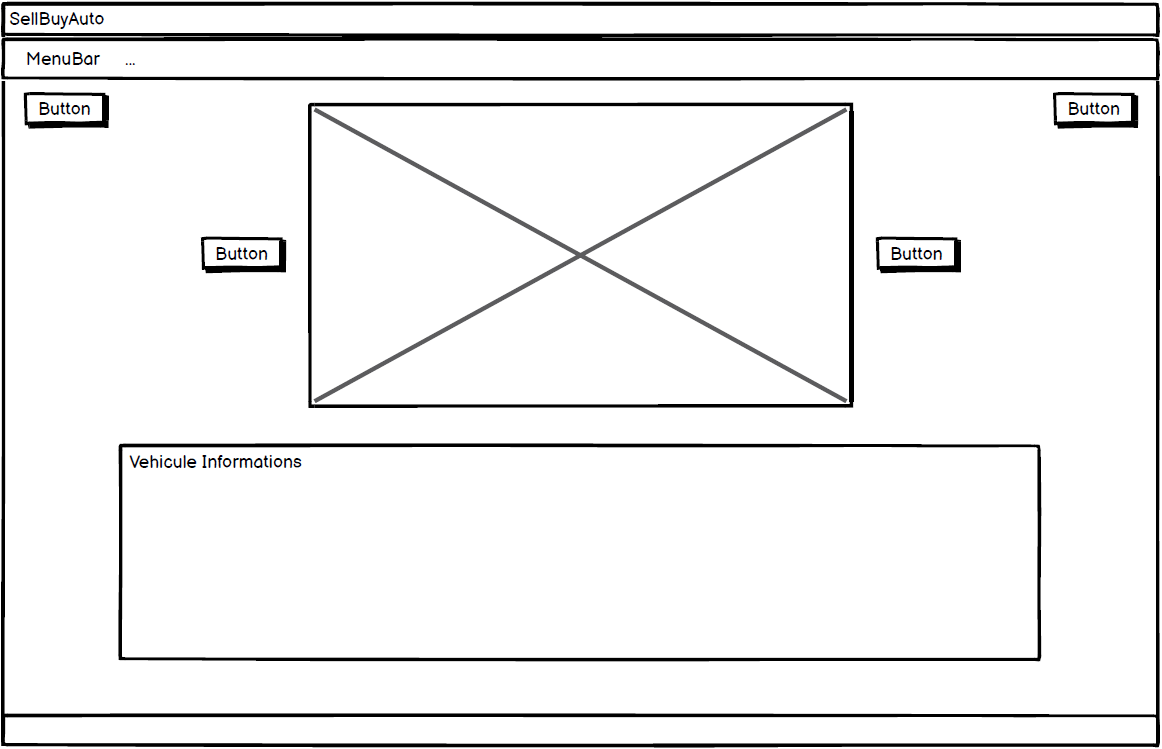


Figure : VehiculeDetailPage\_Zonning

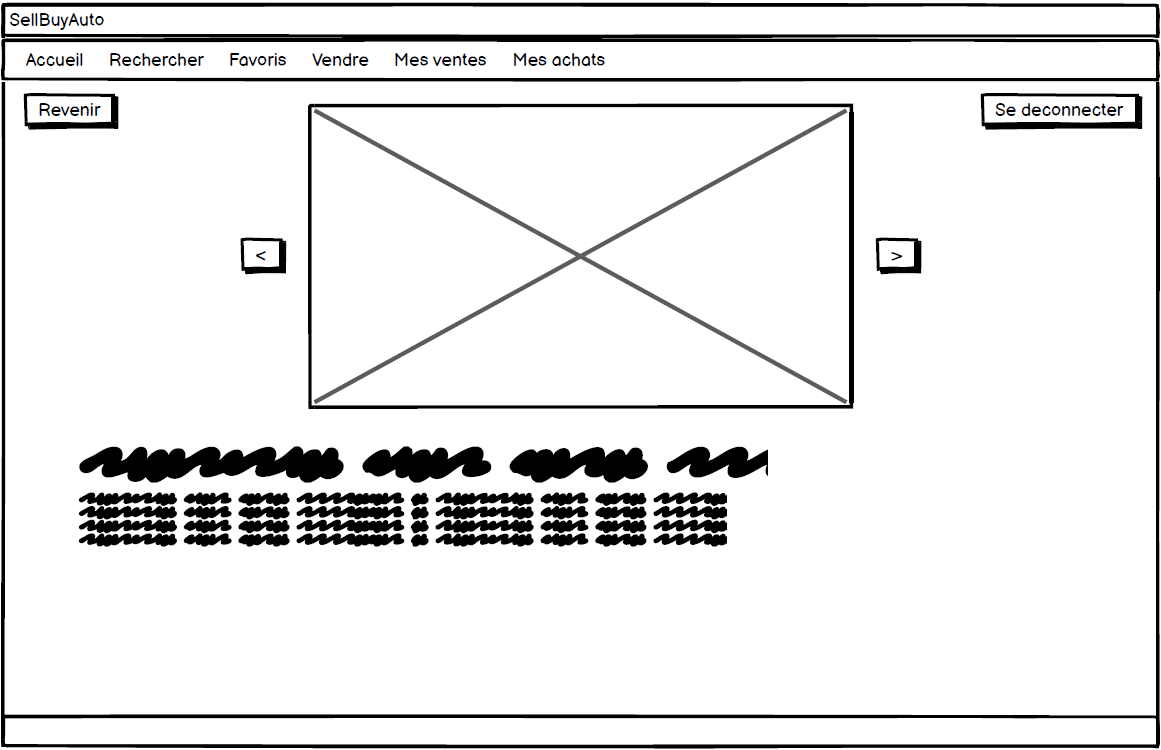


Figure : VehiculeDetailPage\_Wireframe

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.

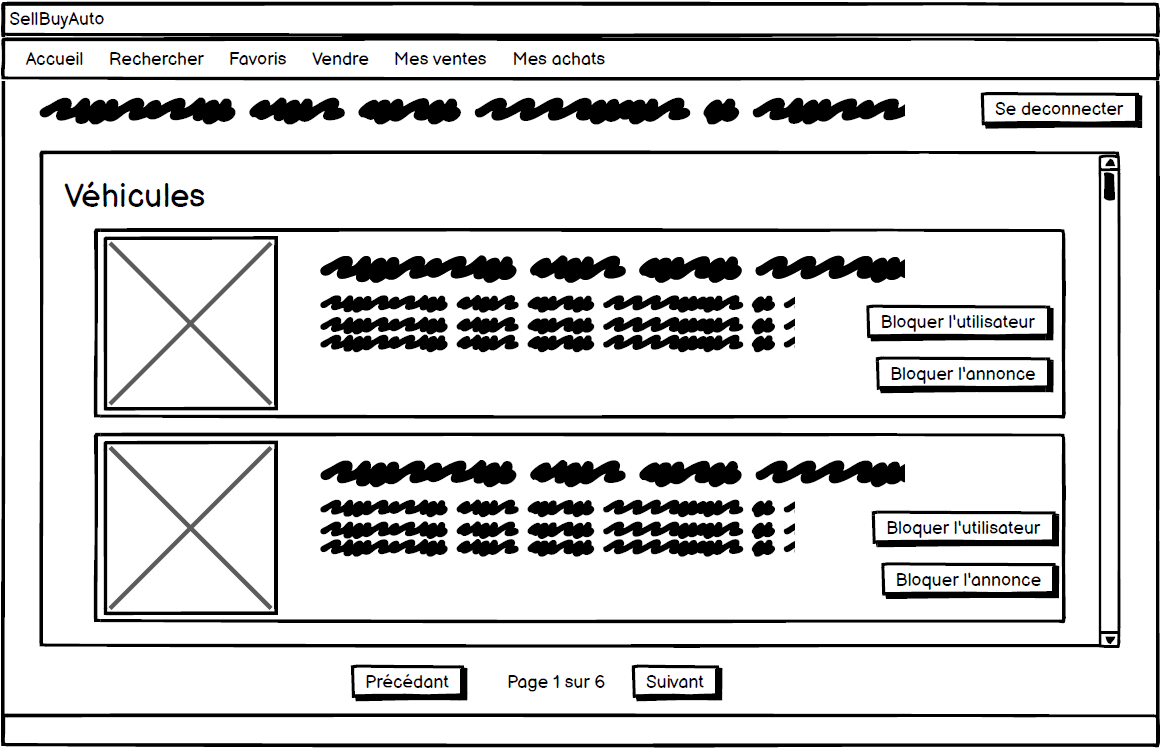


Figure : VehiculesSearchPageInAdmin\_Wireframe

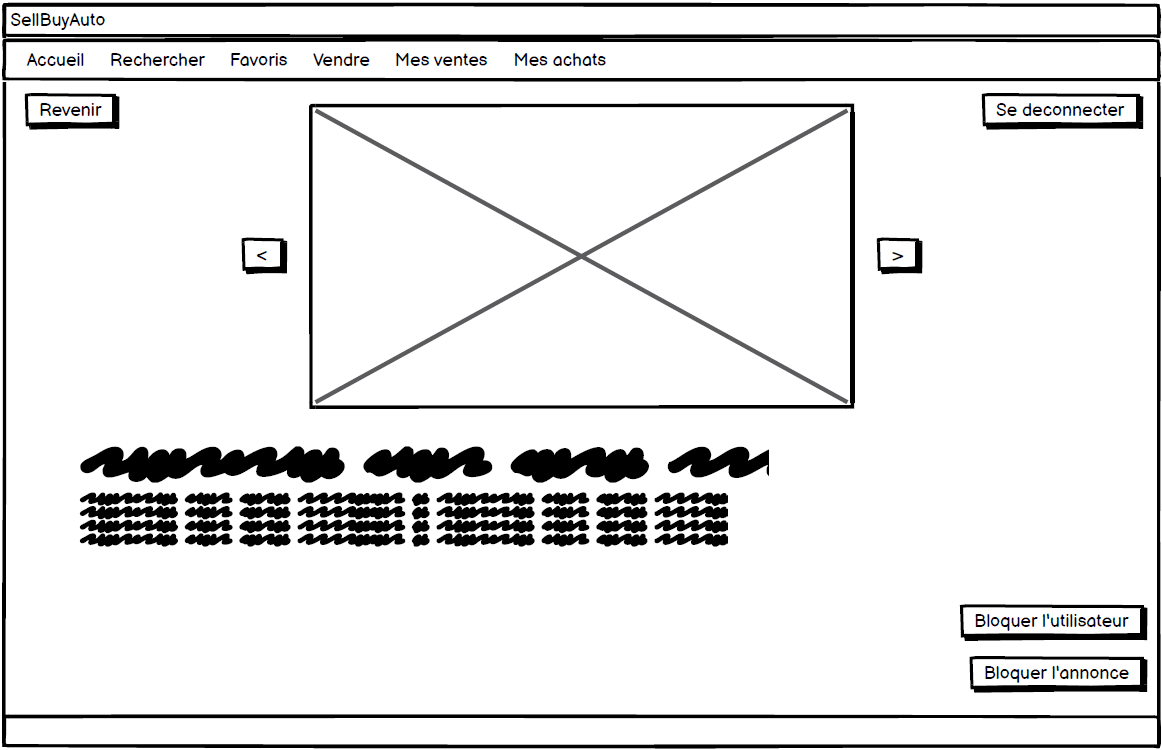


Figure : VehiculeDetailPageInAdmin\_Wireframe

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

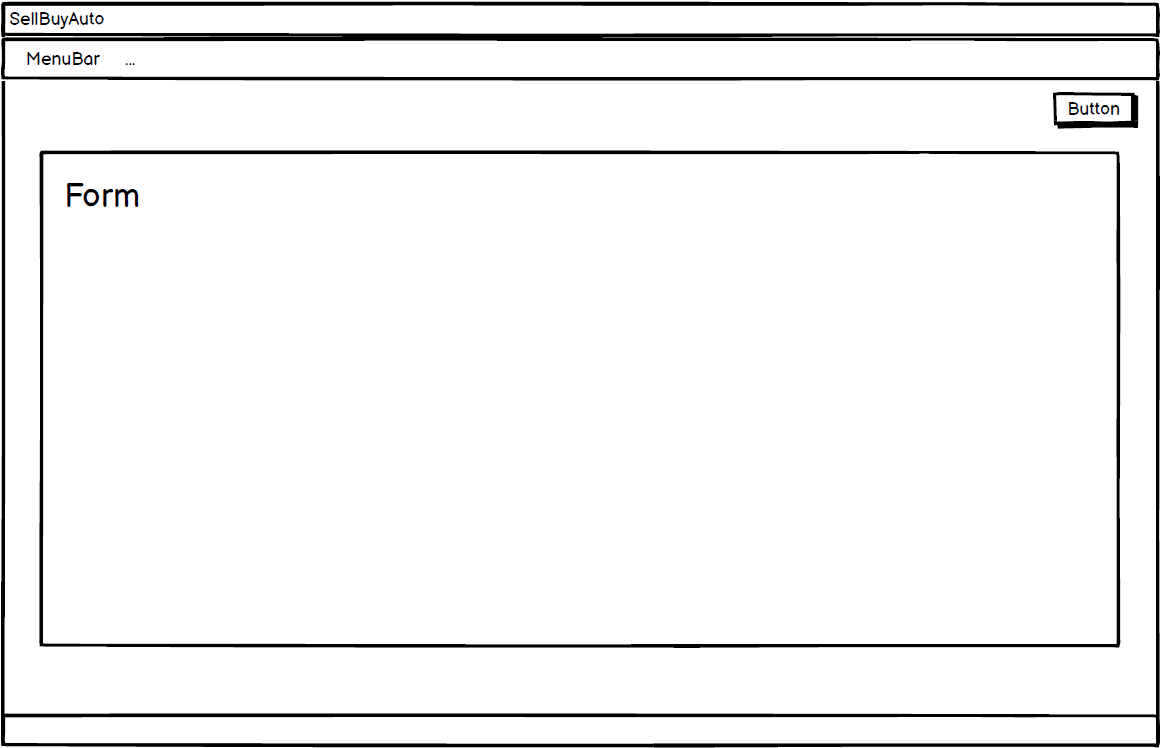


Figure : AddModifyPage\_Zonning

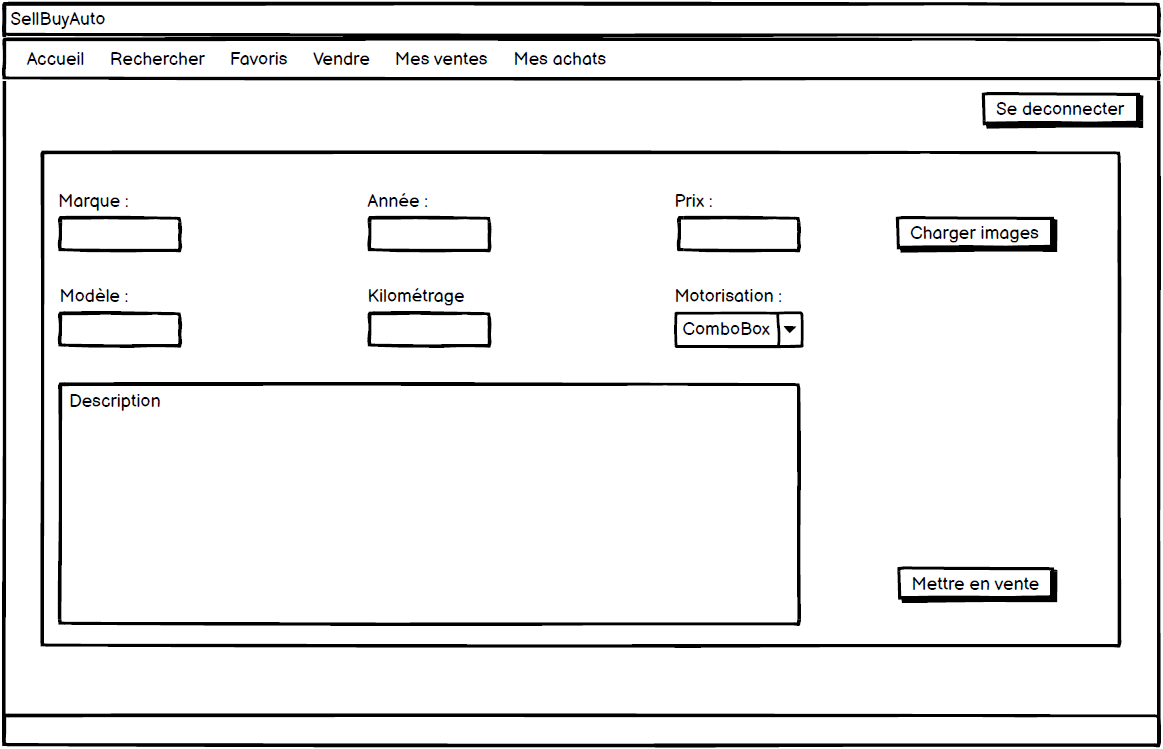


Figure : AddModifyPage\_Wireframe

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.

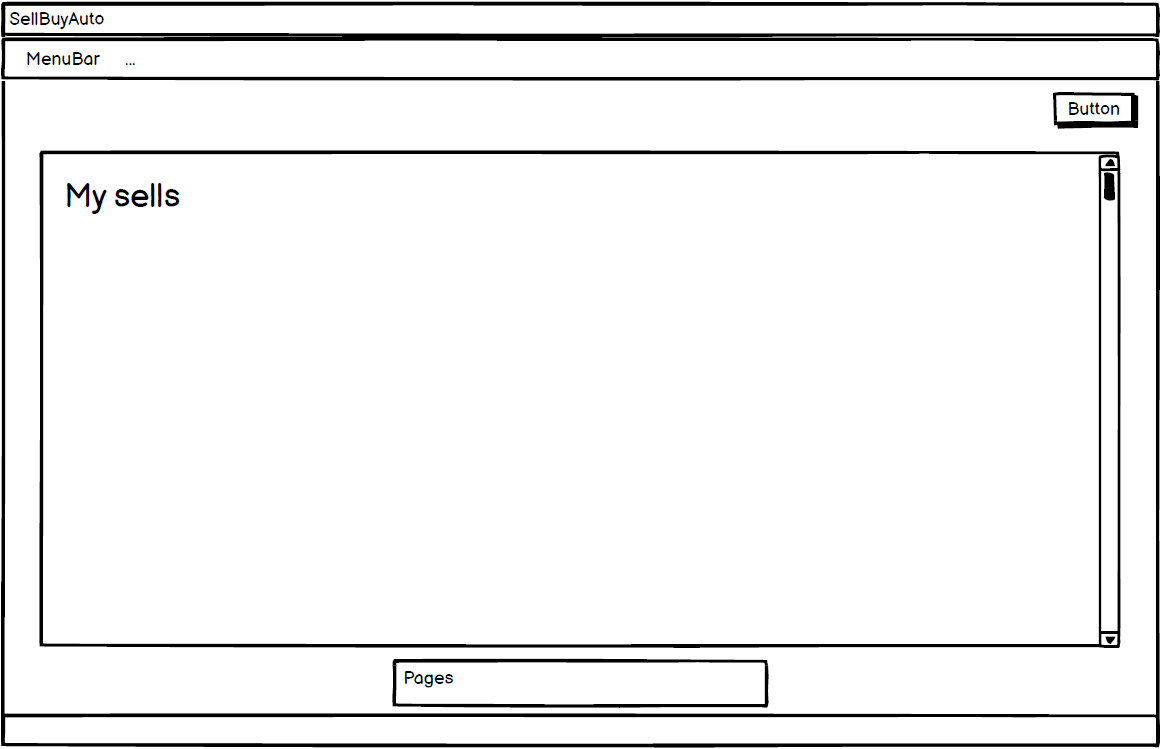


Figure : MySellsPage\_Zonning



Figure : MySellsPage\_Wireframe

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

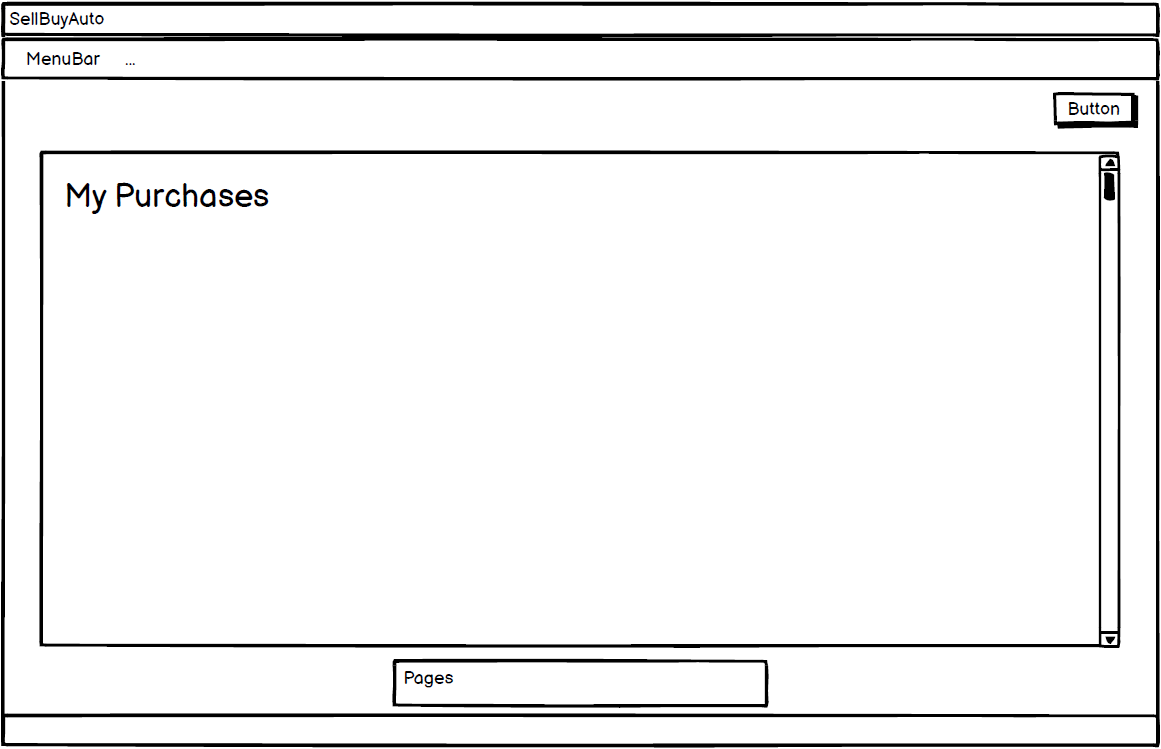


Figure : MyPurchasesPage\_Zonning

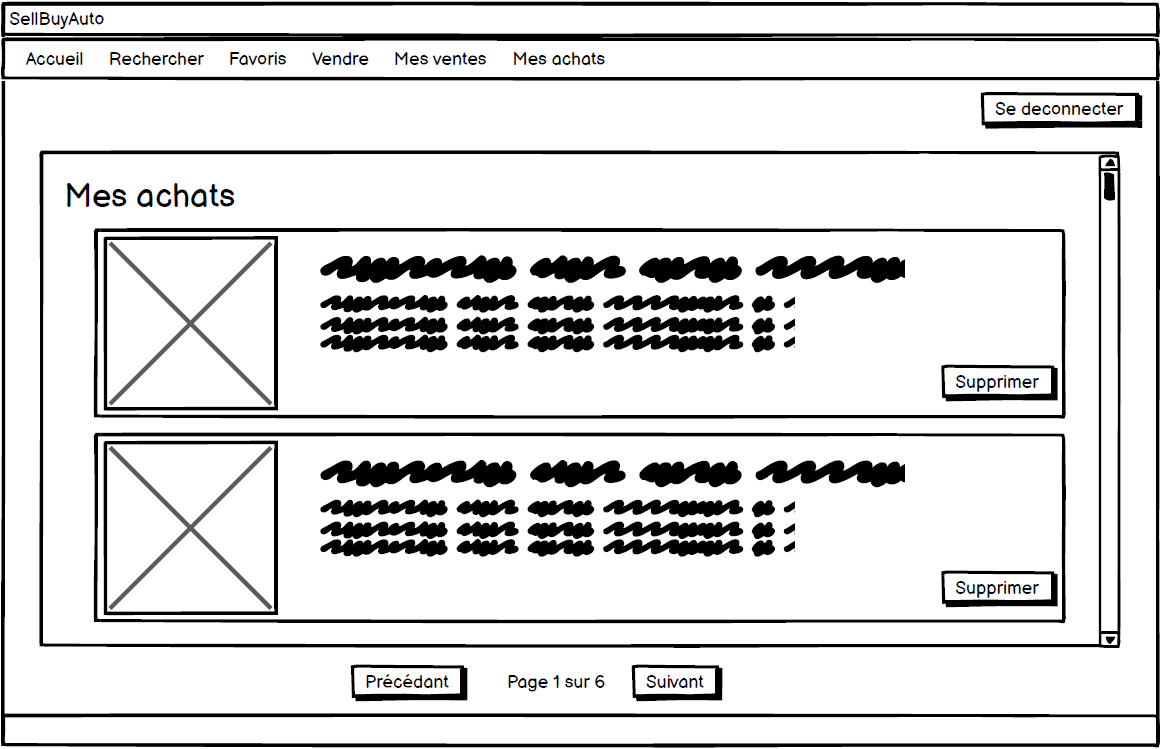


Figure : MyPurchasesPage\_Wireframe

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

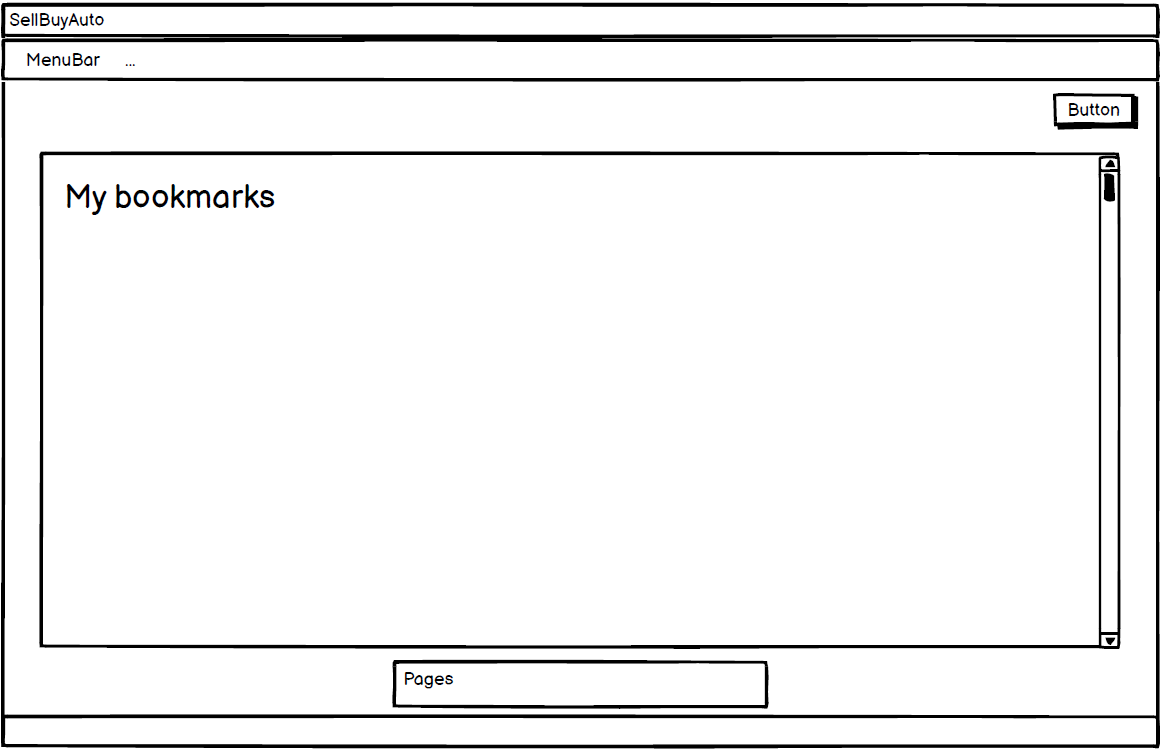


Figure : MyBookmarksPage\_Zonning

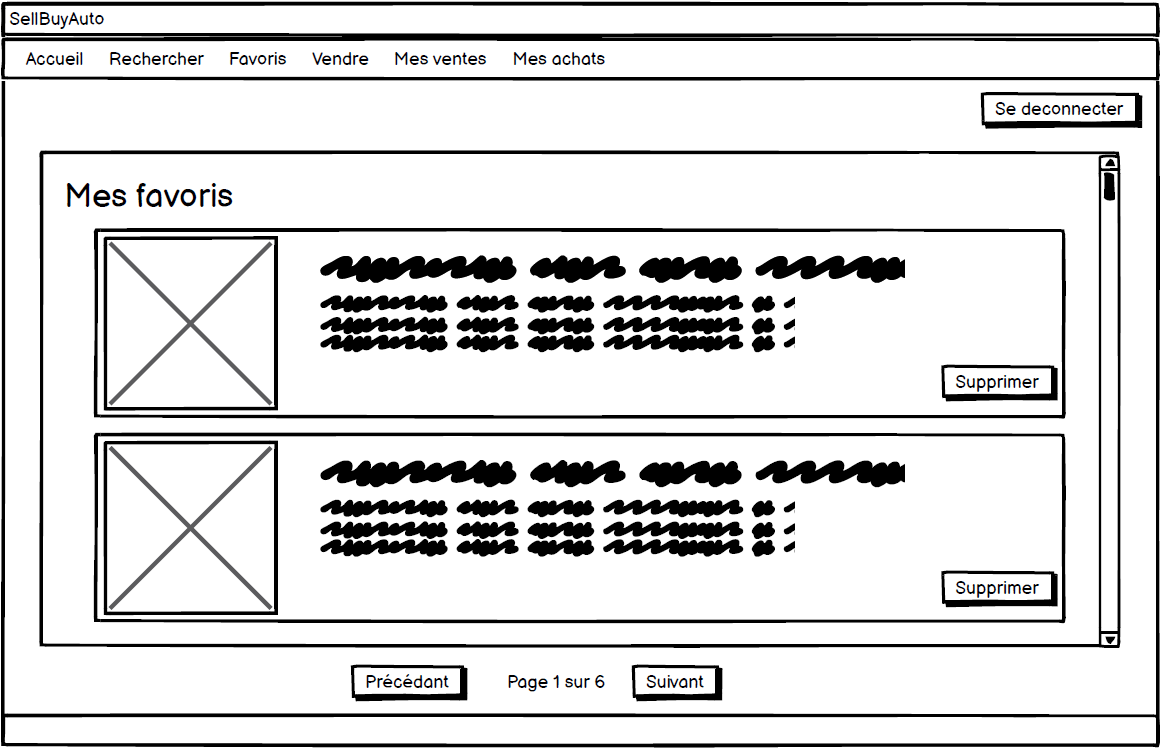


Figure : MyBookmarksPage\_Wireframe

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

### 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 de connexion 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 et un message de succès s’affiche |
|  |  | L'utilisateur n'est pas connecté à son compte | L’utilisateur est redirigé vers la page de connexion |
| Consulter les annonces favorites | Cliquer sur le bouton « Favoris » dans le menu | L'utilisateur est connecté | La liste des annonces favorites de l'utilisateur s’affiche |
|  |  | L'utilisateur n'est pas connecté | L’utilisateur est redirigé vers la page de connexion |
| Consulter ses ventes | Cliquer sur le bouton « Mes ventes » dans le menu | L'utilisateur est connecté | La liste des annonces mises en vente par l'utilisateur s’affiche |
|  |  | L'utilisateur n'est pas connecté | L’utilisateur est redirigé vers la page de connexion |
| Consulter ses achats | Cliquer sur le bouton « Mes achats » dans le menu | L'utilisateur est connecté | La liste des annonces achetées par l'utilisateur s’affiche |
|  |  | L'utilisateur n'est pas connecté | L’utilisateur est redirigé vers la page de connexion |
| Bloquer un utilisateur | 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 |
| 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 : Cas d'utilisation et scénarios

## 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.

## Planification

### 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 débuter le développement et j’aimerais également que ma base de données soit créée.

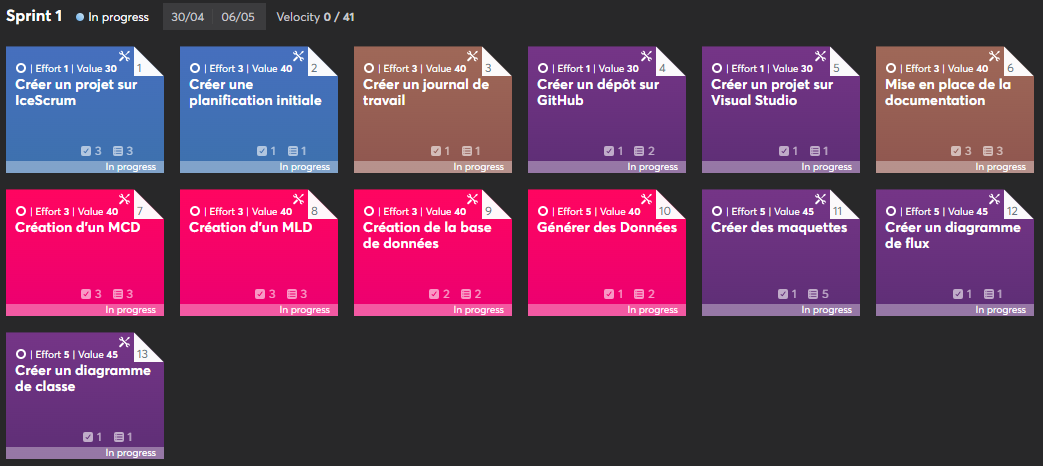


Figure 27 : Stories du sprint 1

### 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.

### 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.

### 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.

# Conception

## 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.

## 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.

## MLD

Figure : 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.

## Diagramme de classes

Figure : 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.

## Diagramme de flux

Figure : Diagramme de flux

# Réalisation

## 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.

### Sprint 1

### Sprint 2

### Sprint 3

### Sprint 4

## Description des tests effectués

## Test de l'Application par un Utilisateur Réel

## Erreurs restantes

# Conclusion

# Annexes

## Résumé du rapport du Pré-TPI

## Glossaire

## Sources – Bibliographie

## Journal de Travail

## Code du script de génération de données