

# Tpi

## « Caves »



## Table des matières

1 Introduction.....	3
1.1 Cadre, description et motivation .....	3
1.2 Organisation.....	3
1.3 Objectifs .....	3
1.3.1 Gestion des publicités conviviale : .....	3
1.3.2 Gestion sécurisée et flexible des informations utilisateur :.....	4
1.3.3 Affichage et gestion détaillés des annonces : .....	4
1.3.4 Visualisation et filtrage anonymes des annonces : .....	4
1.3.5 Facilité de communication et de messagerie :.....	4
1.3.6 Sécurité et protection des données : .....	4
2 Analyse .....	6
2.1 Fonctions prévues .....	6
2.2 Méthode de projet.....	6
2.3 Planification .....	7
2.4 Maquettes.....	10
2.5 User Cases .....	18
2.6 Stratégie de test.....	21
3 Implémentation.....	22
3.1 Choix techniques.....	22
3.2 Modèle conceptuel de données .....	24
3.3 Modèle Logique de données .....	25
3.4 Diagramme de Flux .....	27
3.5 Points techniques spécifiques.....	29
3.5.1 Enregistre les images .....	29
3.5 Bugs.....	31
4 Conclusions.....	31
4.1 Objectifs atteints.....	31
4.2 Objectifs non-atteints .....	31
4.3 Suites possibles au pour le projet .....	31
5 Annexes .....	31
5.1 Journal de travail.....	31
6 SOURCES.....	31

# 1 Introduction

## 1.1 Cadre, description et motivation

Cette application web que nous avons décidé de réaliser a été conçue comme un point de rencontre entre acheteurs et vendeurs. Étant donné que chaque acheteur et vendeur est un utilisateur de l'application, ils seront classés comme utilisateurs et pourront exécuter des fonctions telles que l'ajout d'annonces, l'affichage d'annonces, la saisie et la modification d'informations personnelles et la modification des annonces qu'ils ont placées.

## 1.2 Organisation

Chef de projet : Mr. Pascal BENZONANA, [pascal.benzonana@eduvaud.ch](mailto:pascal.benzonana@eduvaud.ch)  
Expert 1 : Mr. Ernesto MONTEMAYOR, [ernesto@bati-technologie.ch](mailto:ernesto@bati-technologie.ch)  
Expert 2 : Mr. Grégory CHARMIER, [gregory.charmier@eduvaud.ch](mailto:gregory.charmier@eduvaud.ch)  
  
Elève 1 : Yasin Salih Akyüz, [yasin.akyuz@eduvaud.ch](mailto:yasin.akyuz@eduvaud.ch)

	Yasin Salih Akyüz
Partie administration	X
Partie BDD	X
Partie Backend	X
Partie Frontend	X
Partie Panier	X
Partie Carte	X
Mise en page générale	X

## 1.3 Objectifs

- L'utilisateur doit s'inscrire avec succès en saisissant ses informations personnelles.
- L'utilisateur peut se connecter avec les informations qu'il a saisies lors de l'inscription.
- L'utilisateur doit pouvoir saisir une annonce sur sa propre page.
- Les informations personnelles et les annonces doivent être téléchargées dans la base de données et peuvent être modifiées par l'utilisateur.
- Les utilisateurs doivent pouvoir voir l'annonce.
- L'utilisateur peut rechercher des annonces depuis la section de recherche.
- L'utilisateur connecté peut filtrer les annonces selon certains critères.
- Les annonces sont affichées sur la carte selon leurs adresses, et l'utilisateur connecté est affiché différemment sur la carte.
- Les annonces sur la carte sont également affectées par les options de recherche et de filtrage.
- L'utilisateur peut ajouter le produit qu'il souhaite au panier et le produit ajouté au panier sera retiré du stock pendant un certain temps.
- Les utilisateurs peuvent s'envoyer des messages après s'être connectés.

### 1.3.1 Gestion des publicités conviviale :

Les utilisateurs doivent pouvoir créer et publier facilement des publicités à partir de leur propre profil. Lors de la création d'une annonce, les utilisateurs doivent disposer d'options d'édition de texte enrichi

(par exemple, gras, italique, listes, titres, etc.).

#### 1.3.2 Gestion sécurisée et flexible des informations utilisateur :

Les informations personnelles des utilisateurs (nom, e-mail, mot de passe, etc.) doivent être stockées en toute sécurité et les utilisateurs doivent pouvoir mettre à jour ces informations à tout moment.

Lors des mises à jour de mots de passe, les anciens mots de passe doivent être protégés et les nouveaux mots de passe doivent être hachés et stockés en toute sécurité.

#### 1.3.3 Affichage et gestion détaillés des annonces :

Les utilisateurs doivent disposer d'une interface qui permettra une visualisation détaillée des publicités publiées. Les informations telles que la description du produit, le prix, les images et les coordonnées doivent être accessibles sur les pages de détails de l'annonce.

Les utilisateurs doivent pouvoir gérer leurs propres publications et les modifier ou les supprimer à tout moment.

#### 1.3.4 Visualisation et filtrage anonymes des annonces :

Les utilisateurs doivent pouvoir consulter les publications sans se connecter. Ils doivent cependant se connecter pour poster ou communiquer.

Diverses options doivent être proposées pour faciliter le filtrage des annonces, par exemple la fourchette de prix, la catégorie, l'emplacement, etc.

#### 1.3.5 Facilité de communication et de messagerie :

Un système de messagerie doit être prévu entre les utilisateurs pour communiquer avec les annonceurs. Ce système devrait permettre aux utilisateurs de communiquer directement depuis la page de détail de l'annonce.

#### 1.3.6 Sécurité et protection des données :

Des mesures appropriées doivent être prises pour la sécurité des informations et des publicités des utilisateurs, et la sécurité des bases de données doit être assurée.

Il est particulièrement important que les informations sensibles (par exemple les mots de passe) soient stockées et traitées correctement.

#### 1.3.7 Panier d'achat

Les utilisateurs pourront ajouter des produits au panier grâce au bouton « Ajouter au panier » qui devient visible une fois connecté. Dans ce cas, le produit peut être affiché dans le panier, la quantité peut être sélectionnée en fonction de l'état du stock et le prix total changera à mesure que le nombre de produits dans le panier augmentera.

[illegible]

## 2 Analyse

### 2.1 Fonctions prévues

Les principales fonctions prévues dans ce projet sont décrites ci-dessous :

#### Gestion des utilisateurs :

Les utilisateurs peuvent s'inscrire et se connecter au système en créant un compte. Chaque utilisateur se voit attribuer un nom d'utilisateur et un mot de passe. Les utilisateurs peuvent modifier leurs comptes, les supprimer et gérer leurs produits, le cas échéant.

De plus, si l'utilisateur est connecté, il pourra se voir sur la carte, utiliser la page de détails de l'annonce et les options de filtrage.

#### Gestion des produits :

Les utilisateurs peuvent répertorier, modifier et supprimer leurs propres produits. Pour chaque produit, ils peuvent ajouter un titre, une description, un prix et des images si disponibles.

#### Options de filtrage et de recherche

Chaque annonce que les utilisateurs se connectent et ajoute à la base de données peut être filtrée en fonction de leurs préférences personnelles après avoir été affichée sur la carte et la page. Bien que l'option de recherche soit disponible pour chaque utilisateur, l'option de filtrage requise pour la recherche détaillée sera proposée comme fonctionnalité supplémentaire pour l'utilisateur enregistré et connecté.

#### Intégration de la page d'accueil et de la carte :

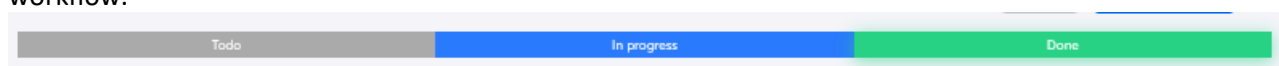
Les produits ajoutés par les utilisateurs sont répertoriés sur la page d'accueil. De plus, grâce à l'intégration cartographique, les utilisateurs peuvent voir les produits à proximité sur la carte en fonction de leur emplacement.

#### Système de messagerie :

Les utilisateurs peuvent utiliser un système de messagerie pour contacter les annonceurs. Il existe un formulaire de contact pour contacter l'annonceur sur la page de détail de l'annonce.

### 2.2 Méthode de projet

Le projet sera géré selon la méthode **Agile**, en utilisant une approche combinée de Scrum et de Kanban. Une approche basée sur le sprint, typique de Scrum, sera adoptée pour optimiser le processus de développement du projet. Chaque sprint visera à atteindre des objectifs fixés sur une période de temps spécifique. Parallèlement, nous intégrerons la méthode **Kanban** pour améliorer la visibilité et la fluidité du workflow.



Ce système Kanban consistera à utiliser un tableau Kanban pour visualiser toutes les tâches du projet, les classer en différentes catégories (par exemple, à faire, en cours, terminé) et limiter le travail en cours pour assurer une gestion continue et efficace des ressources. Cette combinaison des deux méthodes permettra une flexibilité maximale et une amélioration continue tout au long du cycle de développement du projet.

## 2.3 Planification

Des outils appropriés seront utilisés pour planifier et suivre le projet.

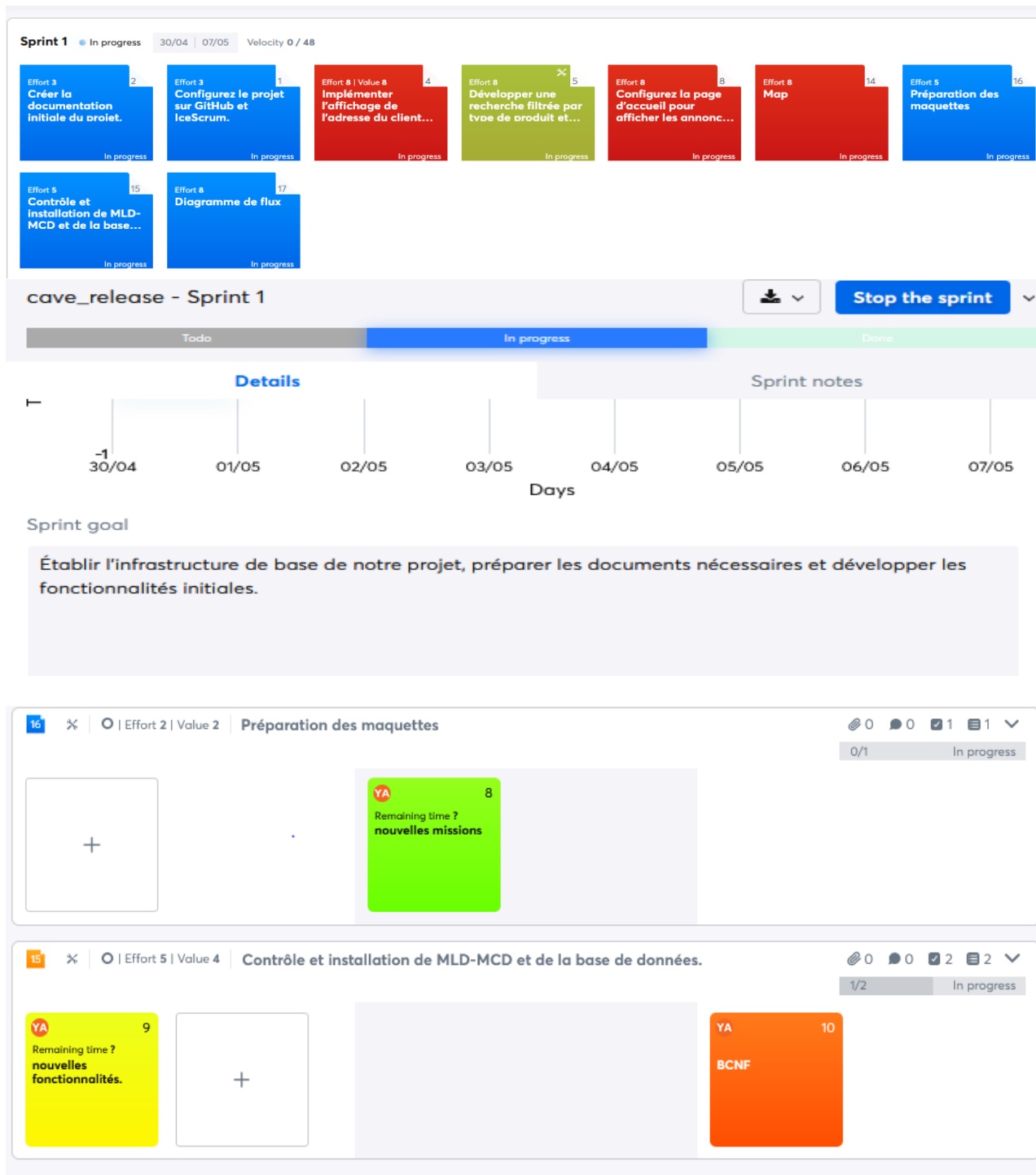
La répartition du travail et le calendrier seront déterminés et suivis tout au long de la gestion du projet. De plus, des étapes et des sprints importants seront déterminés et des efforts seront déployés pour atteindre ces objectifs.

J'utilise "Ice Scrum" pour suivre et planifier le projet. Pendant que nous suivons l'avancement du projet avec des fonctionnalités telles que le sprint, la tâche, le test, nous rendons le suivi du projet plus détaillé.

- Sprint 1 (30.04-07.05)

- Dans la phase de développement du site web, les travaux ont commencé sur la préparation des documents, la réalisation du projet, et l'établissement de la base de données, parmi d'autres tâches nécessaires et supportives.
- Pour contrôler, organiser et suivre facilement la progression du projet, un dossier local et un dossier GitHub ont été créés, et une structure de suivi avec iceScrum a été mise en place.
- Pendant le sprint-1, un test-task a été ajouté à chaque histoire pour faciliter le suivi, des mises à jour ont été faites sur le tableau des tâches et un objectif de sprint a été créé.
- Des maquettes suggérant l'apparence des pages web ont été préparées en utilisant le programme Balsamiq.
- À l'aide de Draw.io, un tableau "MCD" (Modèle Conceptuel de Données) a été détaillé, travaillant en profondeur sur les relations de référence entre les tables prévues pour la base de données.
- Le tableau "MLD" (Modèle Logique de Données) a été finalisé sur "MySQL Workbench", avec l'ajout de détails tels que clé primaire, clé étrangère, null, etc., et les liens entre les tables ont été revus.
- Les opérations de base de données ont été achevées avec l'utilisation de "HeidiSQL", les tables ont été établies, et des opérations d'INSERT et de SELECT de test ont été effectuées pour vérifier le fonctionnement des tables et leurs interconnexions.
- La page d'accueil a été organisée pour être utilisable pour les opérations de backend.
- Mapbox GLJS a été choisi pour l'utilisation de la carte et ajouté à la page d'accueil pour améliorer l'expérience utilisateur.
- Un diagramme de flux a été préparé avec le programme Draw.io et sera mis à jour tout au long des sprints.
- Des options de filtrage et de recherche ont été ajoutées au projet, offrant de nouvelles fonctionnalités aux utilisateurs.





- Sprint 2 (08.05-16.05)

Au cours du deuxième sprint, des efforts seront déployés pour augmenter les fonctionnalités visuelles et utilisables de l'application par les utilisateurs et pour garantir que les instructions soient explicatives pour l'utilisateur.

De plus, augmenter l'expérience de vente de l'utilisateur et rendre les fonctions du panier fonctionnelles seront un objectif distinct et important.



## Sprint 2

● Todo

08/05 | 16/05

Planned velocity 26

Effort 5  
**Concevoir une interface de base en suivant les directiv...**  
3  
Planned

Effort 8 | Value 8  
**Implémenter une fonctionnalité d'affichaae des...**  
6  
Planned

Effort 13 | Value 11  
**Mettre en place un système de réservation de...**  
10  
Planned

- Sprint 3 (17.05-23.05)

## Sprint 3

● Todo

17/05 | 23/05

Planned velocity 18

Effort 13  
**Mettre en place un système de messaaerie entre...**  
7  
Planned

Effort 5  
**Commencez à tester les fonctionnalités développées iusau'...**  
9  
Planned

- Sprint 4 (24.05-29.05)

## Sprint 4

● Todo

24/05 | 29/05

Planned velocity 21

Effort 8  
**Finaliser les ajustements de l'interface utilisateur.**  
11  
Planned

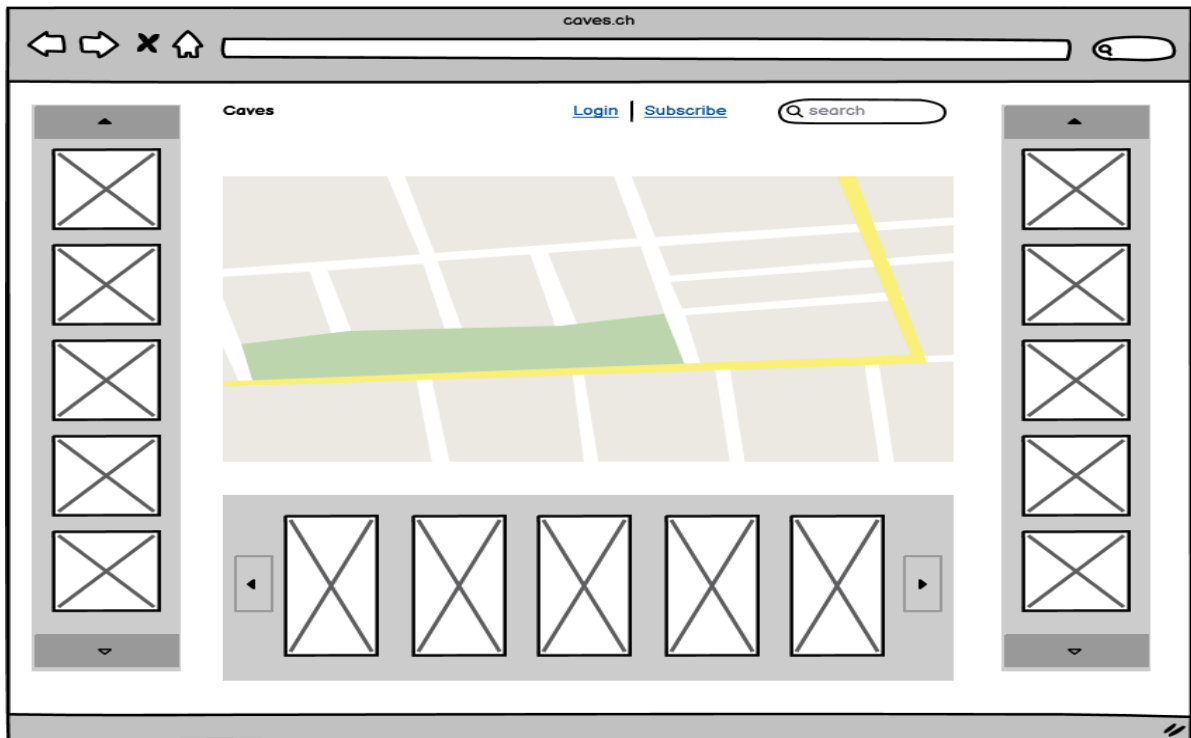
Effort 8  
**Effectuer des tests approfondis, y comoris des tests...**  
12  
Planned

Effort 5  
**Préparer la documentation finale du projet et l...**  
13  
Planned

## 2.4 Maquettes

- Lorsque l'utilisateur ouvre la page web, il sera accueilli par la page « PAGE ACCUEIL » qui apparaîtra comme suit.

Sur cette page, l'utilisateur pourra s'inscrire et se connecter, rechercher des produits. Cependant, bien qu'il puisse voir les annonces, il devra se connecter pour entrer en contact avec l'annonceur.



Dans cette page, si l'utilisateur est connecté ou non, l'utilisateur n'a pas accès aux mêmes actions :

- ⇒ Non-connecté :
  - Login
  - Subscribe
  - Les Annonces
- ⇒ Connecté :
  - Logout
  - Filtrer
  - Messages
  - Mypage
  - Les Annonces

Avant de vous connecter, vous devez vous inscrire en saisissant vos informations d'utilisateur. Lorsque vous cliquez sur le bouton S'abonner, une fenêtre contextuelle s'ouvrira et l'utilisateur saisira et enregistrera ses informations. Ensuite, en se connectant, il pourra modifier ses informations personnelles, publier une annonce et contacter d'autres annonceurs.

Lorsque l'utilisateur souhaite s'inscrire, il cliquera sur le bouton "Subscribe", et à ce moment-là, une page ou une fenêtre pop-up s'ouvrira pour entrer les informations de l'utilisateur.

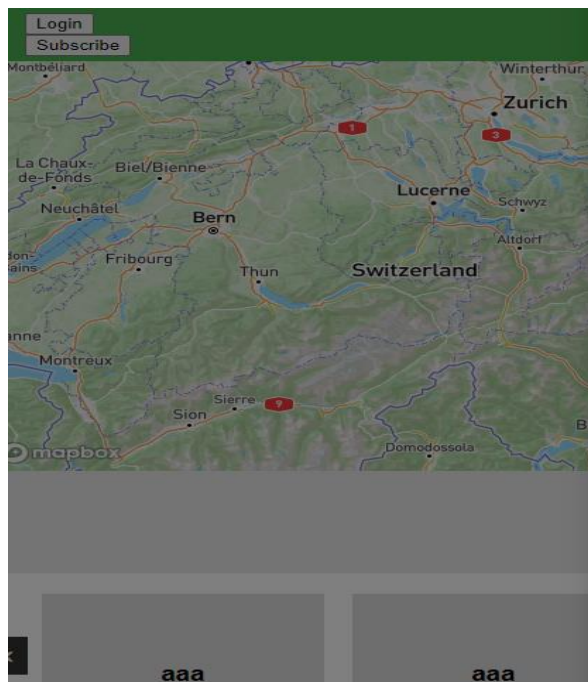
Lors de la saisie des informations, l'utilisateur verra un écran d'avertissement lui indiquant de saisir

son adresse e-mail et son numéro de téléphone conformément au type de données défini dans la base de données. Ainsi, il pourra compléter le processus d'inscription avec succès.

The screenshot shows a web browser window with the address bar displaying 'https://'. The page title is 'A Web Page'. The main content area has a header with 'Caves' on the left and a 'Login' button on the right. Below the header is a form with the following fields:

- Nom de l'entreprise ( facultatif) : [Text input]
- Nom : [Text input]
- Prénom : [Text input]
- e mail : [Text input]
- Phone : [Text input]
- Rue et N° : [Text input]
- Code Postal : [Text input]
- Ville : [Text input]
- Canton : [Text input]
- Password : [Text input]
- Confirm Password : [Text input]

At the bottom left of the form is a 'Subscribe' button.



## Subscribe

Nom de l'entreprise (facultatif)

Name

Firstname

E mail

Numéro de téléphone

Adresse

Numéro du bâtiment

Code postal

Ville

Canton

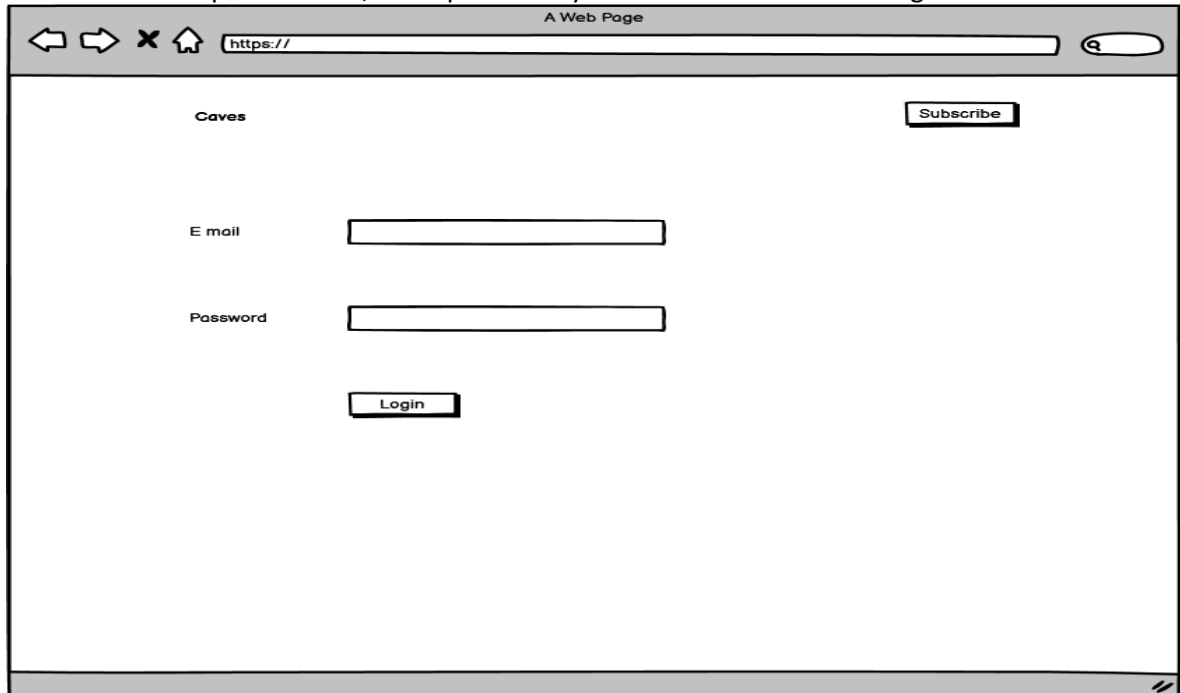
Le mot de passe

Confirmez le mot de passe

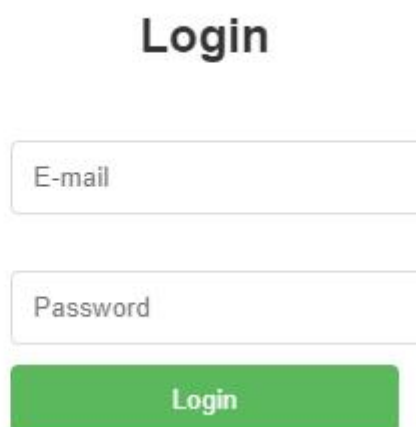
Subscribe

- Après avoir terminé le processus d'inscription, l'utilisateur pourra se connecter à sa page utilisateur en utilisant l'adresse e-mail enregistrée et le mot de passe choisi.

Il aura alors la possibilité de publier des annonces, de modifier les annonces publiées, de modifier ses informations personnelles, ainsi que d'envoyer et de recevoir des messages.



A screenshot of a web browser window titled "A Web Page". The address bar shows "https://". The page content includes a "Caves" header with a "Subscribe" button. Below this, there are two input fields labeled "E mail" and "Password", followed by a "Login" button.

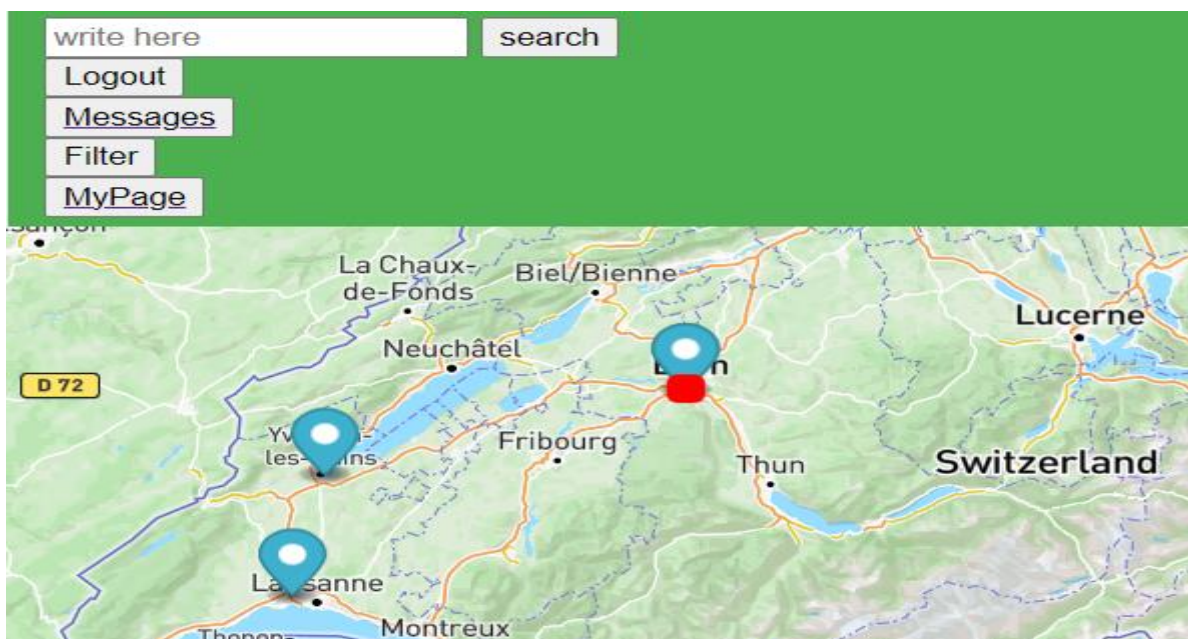
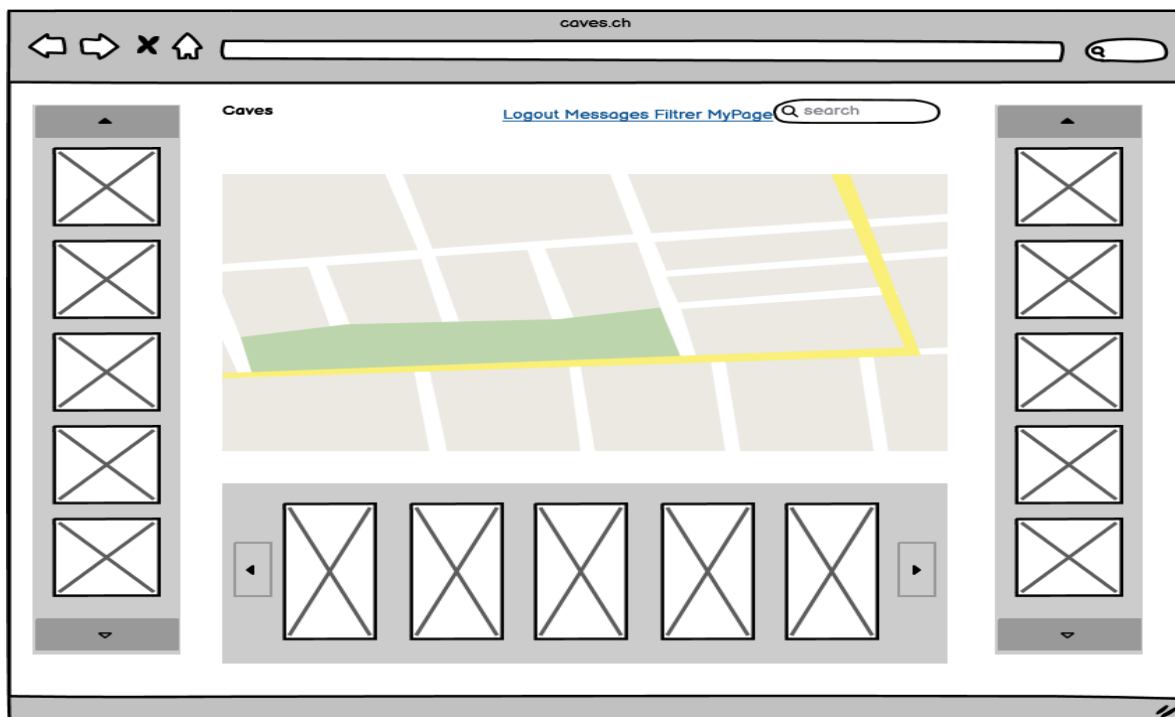



A login form with a green header containing "Login" and "Subscribe" buttons. Below the header, there are two input fields labeled "E-mail" and "Password", followed by a green "Login" button.

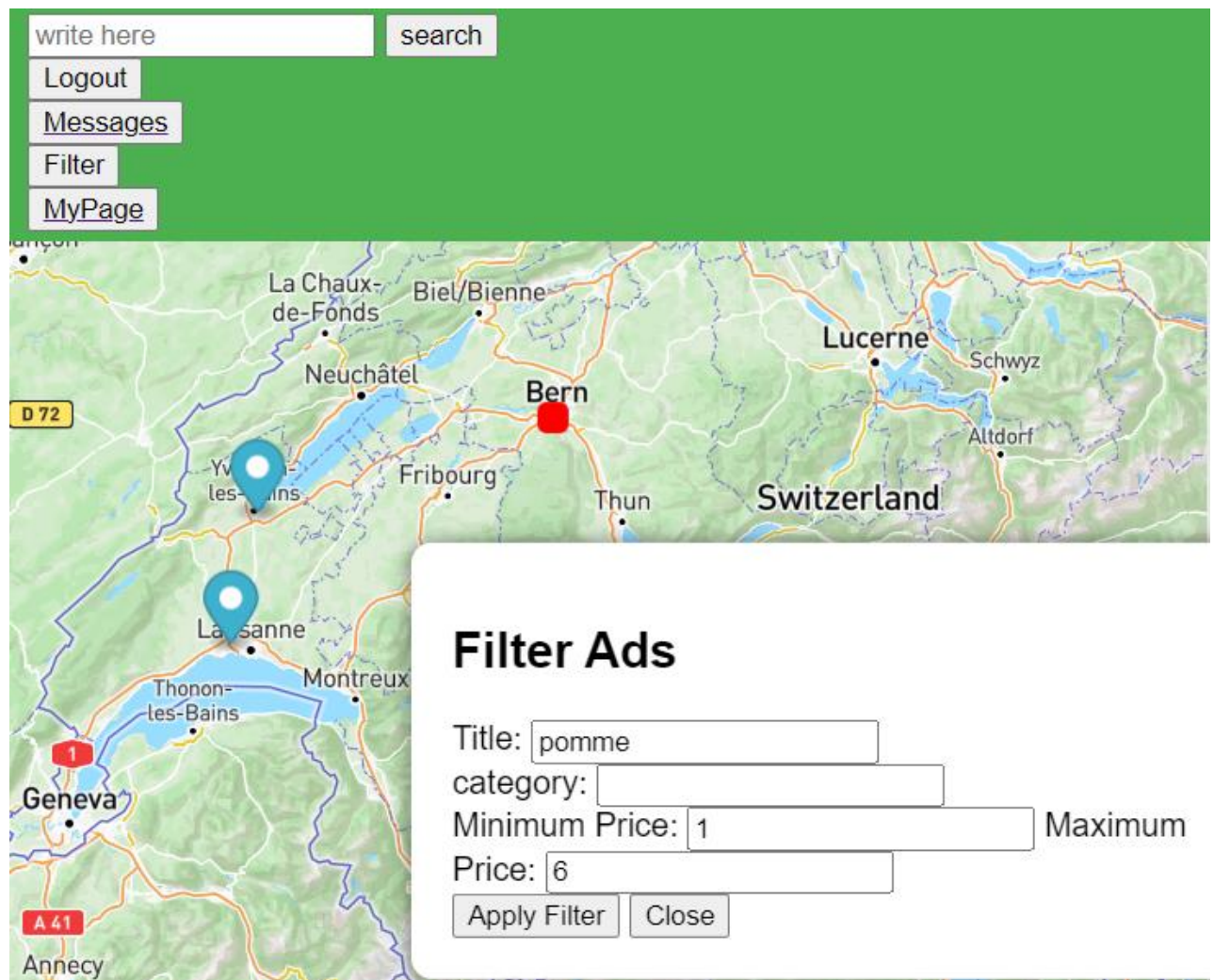
- Après l'enregistrement de l'utilisateur, les boutons sur la page d'accueil changeront pour offrir différentes fonctionnalités.

En cliquant sur le bouton MyPage, l'utilisateur sera redirigé vers sa page personnelle. En cliquant sur le bouton de déconnexion, il pourra se déconnecter.

Le bouton Filtrer permettra de réaliser des recherches détaillées, et le bouton messages permettra à l'utilisateur de voir ses messages.



L'utilisateur pourra également effectuer les opérations de filtrage qu'il jugera nécessaires en sélectionnant l'option de filtrage sur cette page. Dans l'exemple, le filtrage sous l'en-tête « titre » est affiché. À ce stade, lorsque l'on clique sur le bouton « Appliquer le filtre », les publicités affichées sur cette page s'affichent en fonction du résultat du filtrage.



```
Starting to clear markers. Total markers before clearing: 1 accueil.js:42
All markers removed, markers array reset. accueil.js:45
Ads fetched successfully: ▶ (2) [{...}, {...}] accueil.js:59
Ad Data: accueil.js:75
  ▶ {id: '76', title: 'pomme', situation: 'pommeysa', creation_date: '2024-05-07 10:46:43', street: "Rue de L'industri
    e", ...}
Ad Data: accueil.js:75
  ▶ {id: '77', title: 'pomme', situation: 'good', creation_date: '2024-05-07 11:07:15', street: 'Rue de La Mère', ...}
Adding marker for ad: pomme with query: Rue%20de%20l'industrie%2028%2C%201400%20Yverdon-les- accueil.js:92
Bains%2C%20Vaud
Adding marker for ad: pomme with query: Rue%20de%20la%20Mère%2029%2C%201020%20Renens%2C%20Vaud accueil.js:92
Marker added, total markers now: 1 accueil.js:107
Marker added, total markers now: 2 accueil.js:107
```

- La page "MyPage" est une page personnelle où l'utilisateur peut effectuer et organiser ses transactions en détail, unique pour chaque utilisateur.

Sur cette page, l'utilisateur peut modifier toutes ses informations avec des redirections, poster de nouvelles annonces et voir les annonces qu'il a postées.

Cette page, que nous pourrions appeler une page de modification, permettra à l'utilisateur d'interagir de multiples manières dans les étapes futures et restera ouverte à des ajouts. Par exemple, l'option de supprimer le compte de l'utilisateur pourrait être ajoutée à cette page



plus tard.

A Web Page
https://mypage

### Edit Personal Informations

Nom	<input type="text"/>	Prénom	<input type="text"/>
e mail	<input type="text"/>	Phone	<input type="text"/>
Rue et N°	<input type="text"/>	Code Postal	<input type="text"/>
Ville	<input type="text"/>	Canton	<input type="text"/>
Password	<input type="text"/>	<input type="button" value="Save Changes"/>	

### Post an Ad

<input type="text" value="Title"/>	<input type="text" value="Situation"/>
<input type="text" value="Categories"/>	<input type="text" value="product name"/>
<input type="text" value="Price"/>	<input type="text" value="Stock"/>
Images : <input type="button" value="select fichier"/>	<input type="button" value="PostAd"/>

### My Ads

<div>Item One</div>	<div>Item Two</div>	<div>ItemThree</div>	<div>Item Four</div>
---------------------	---------------------	----------------------	----------------------

L'utilisateur pourra à ce stade consulter ses informations personnelles et apporter les modifications souhaitées. Lors de la modification, les anciennes données seront préservées dans les champs laissés vides. Le nouveau mot de passe sera également enregistré dans la base de données sous forme de hachage.

Dans la section de publication d'annonce, il pourra publier une annonce détaillée en saisissant les informations nécessaires. Il pourra également sélectionner plusieurs images et les enregistrer dans la base de données sous forme d'URL.

Il pourra consulter les annonces qu'il a publiées dans la section Liste des annonces et, à l'avenir, effectuer des opérations de modification et de suppression sur ces annonces.

## My Page

### Edit Personal Information

Name	first Name	Company Name	Email	Phone	Street	Building Number	Postal Code
City	Canton	new password	<input type="button" value="Save Changes"/>				

### Post an Ad

Ad Title	situation	fruit
product name: Urin Adi		
price:		
Stock:		
images: <input type="button" value="Sélect. fichiers"/>	Aucun fichier choisi	
<input type="button" value="Post Ad"/>		

### Ads List

mais2

bon



Product Name: mais fruit

Price: 1

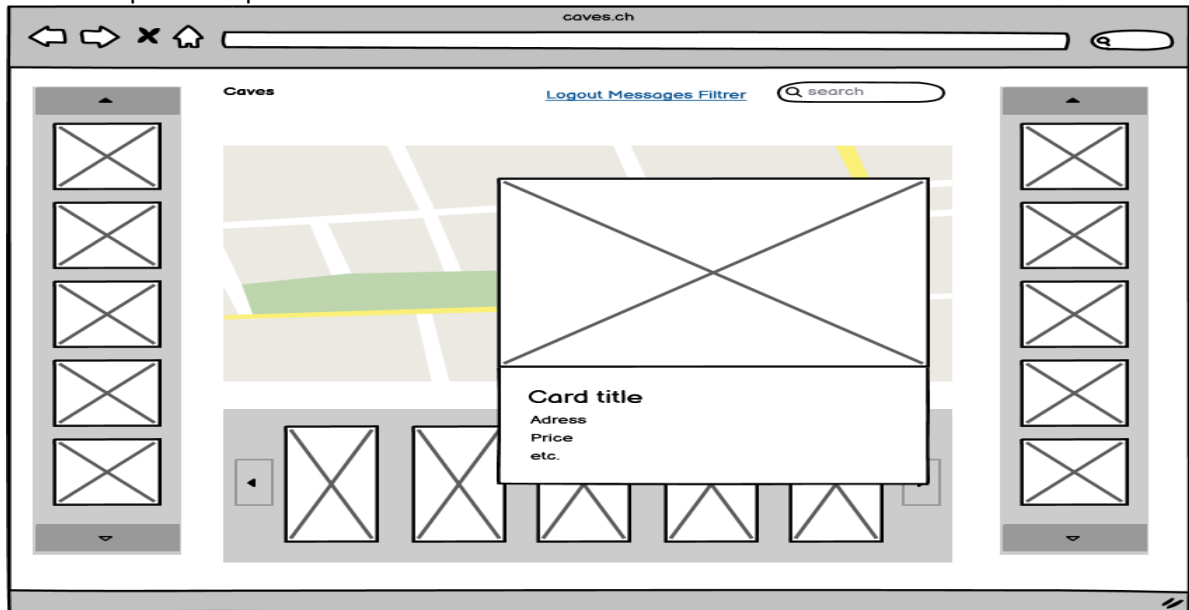
Stock: 4



- Sur la page d'accueil (après la connexion), l'utilisateur peut voir les annonces qui ont été publiées précédemment.

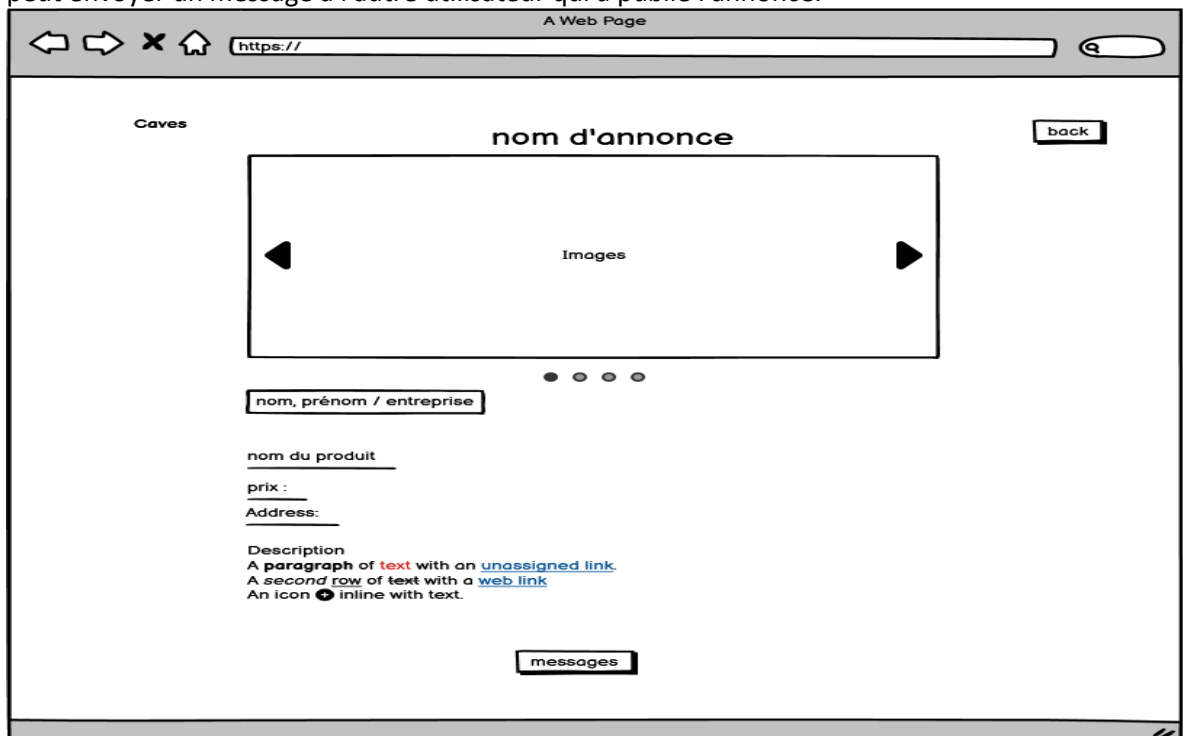
Lorsque l'utilisateur passe la souris sur une annonce, un pop-up apparaîtra, fournissant des informations succinctes sur celle-ci.

Avec ces informations, si l'utilisateur le souhaite, il pourra accéder à une page de détails de l'annonce pour voir plus de détails.



- La page de détails de l'annonce s'ouvre lorsque l'utilisateur clique sur une annonce.

Sur cette page, l'utilisateur peut voir les informations détaillées de l'annonce et, s'il est connecté, peut envoyer un message à l'autre utilisateur qui a publié l'annonce.



- Un utilisateur qui s'est connecté et qui visualise une annonce peut poser directement des questions à l'utilisateur ayant publié l'annonce, permettant ainsi une communication directe entre eux sans avoir à partager leurs informations de contact personnelles dans l'annonce.

Ce système est également une caractéristique importante pour protéger les données personnelles de chaque utilisateur, évitant la nécessité de divulguer des informations de contact personnelles dans les annonces.

A Web Page

https://

Caves

messages

back

une petit information sur l'annonce

prix

expéditeur du message

destinataire du message

Ce produit est-il toujours disponible?

Envoyer

- Après avoir effectué la connexion, un des boutons qui apparaît est le bouton « Mes messages ».

En cliquant sur ce bouton, l'utilisateur peut voir tous les messages qu'il a envoyés et reçus en un seul endroit, et il a également la possibilité de supprimer l'historique des messages qu'il souhaite.

A Web Page

https://

Caves

MESSAGES

une petit information sur l'annonce

Supprimer

- Une fois que l'utilisateur s'est connecté et a accédé à la page de publicité, il verra le bouton Ajouter au panier. Lorsque vous cliquez sur ce bouton, le produit sera ajouté au panier. Lorsque vous cliquez sur le bouton du panier, la page du panier s'ouvrira et les détails du panier seront affichés.

-

## 2.5 User Cases

« Use Case » Les cas d'utilisation (scénarios d'utilisation) mentionnés dans les tableaux définissent les comportements et conditions attendus pour diverses situations d'interaction des utilisateurs avec une plateforme web. Ces informations, fournies sur deux pages, couvrent les fonctions de base que les utilisateurs peuvent exécuter sur le site web et présentent les scénarios et conditions possibles pour chacune de ces fonctions.

Le premier tableau contient les scénarios définis pour la page d'accueil du site, qui est le point de départ de l'accès des utilisateurs. Ces scénarios incluent des processus tels que l'affichage du formulaire d'inscription et la création d'un nouvel utilisateur. Pour chaque scénario, les actions requises (par exemple, cliquer sur un bouton), les conditions nécessaires à leur réalisation et les résultats attendus (succès ou échecs) sont spécifiés.

Le deuxième tableau traite des scénarios après que l'utilisateur se soit connecté au système. Ces scénarios incluent des fonctions telles que la mise à jour d'informations, la publication d'annonces, la fermeture de session et la gestion des messages. Les actions nécessaires, les conditions et les résultats possibles pour chaque scénario sont également listés en détail.

Objectif du Use Case	Action	Conditions	Scénario condition remplie	scénario echec condition
<b>Page principale (non-connecté)</b>				
Visionner des publicités	Cliquez sur les annonces	pas pouvoir contacter l'annonceur sans se connecter	Affichage des informations sur les annonces via la base de données	
Afficher formulaire d'inscription	Click sur bouton "subscription"		Affichage du formulaire d'inscription	
Créer un nouvel utilisateurs	Click sur bouton "Subscribe"	Ne pas avoir de compte avec la meme adresse email	Inscription de l'utilisateur dans la base de donnée	Message d'erreur s'affiche "un compte est déjà lié à cette adresse email"
			Session utilisateurs "ouverte", avec ses données chargées dans le sessionstorage	Avertissement si les informations saisies ne correspondent pas aux types de données. Avertissement si les informations saisies ne correspondent pas aux types de données. Email, postal code, building number.  Avertissement que les deux mots de passe saisis ne correspondent pas.
Fermer le formulaire d'inscription	Cliquez n'importe où sur l'écran		Fermeture du formulaire d'inscription	
Afficher formulaire de login	Click sur bouton "LOGIN"		Affichage du formulaire de login	
Connecter un utilisateurs	Click sur bouton "LOGIN" du formulaire login	Avoir déjà créé un compte	bouton "login" se transforme en bt "logout"	Avertissement "@" si les informations de email sont mal saisies
			un bouton "mypage" apparait	Avertissement donné si toutes les cases ne sont pas remplies
		valeurs email & password correctes	un bouton "messages" apparait	un message d'erreur s'affiche : "vos identifiants sont incorrects"
		Tout les champs du formulaires sont remplis	un bouton "filter" apparait	
Fermer le formulaire de login	Cliquez n'importe où sur l'écran		Fermeture du formulaire de login	

Objectif du Use Case	Action	Conditions	Scénario condition remplie	scénario echec condition
<b>Page principale (connecté)</b>				
Page personnelle Click bt "MyPage"	Click bt "Save Changes"	De nouvelles données ont-elles été saisies conformément aux types de données	Inscription des informations	Avertissement "@" si les informations de email sont mal saisies
			Si aucune information n'est saisie, les anciennes informations sont conservées.	Avertissement si les informations saisies ne correspondent pas aux types de données. Avertissement si les informations saisies ne correspondent pas aux types de données. Email, postal code, building number
	Click bt "Post Ad"	Recevez un avertissement si les statuts de prix et de stock restent vides	La publicité est enregistrée dans la base de données.	Si les informations sur les prix et les stocks sont incomplètes, l'annonce ne sera pas enregistrée dans la base de données.
			Les photos sont enregistrées dans la base de données sous forme d'URL.	
Se déconnecter	click bt "LOGOUT"	Avoir déjà créé un compte et être connecté dessus	bouton "LOGOUT" se transforme en bouton "LOGIN"	Le bouton est invisible
			bouton "MyPage" disparaît	
			bouton "Messages" disparaît	
			bouton "Filter" disparaît	
Page Messages	Click bt "Messsages"	Avoir déjà créé un compte et être connecté dessus	L'utilisateur peut voir et supprimer tous les messages précédents.	
		L'utilisateur peut cliquer sur le bouton Supprimer les messages		

## 2.6 Stratégie de test

L'utilisateur sera confronté à de nombreuses options lors de l'utilisation de l'application.

L'utilisateur devra saisir les informations correctes et agir conformément à notre système établi.

Dans ce contexte, il sera de notre devoir de guider l'utilisateur et d'obtenir des données saines.

Il est nécessaire de tester à la fois en termes de guidage de l'utilisateur et de fonctionnalité de l'application.

Vous pouvez trouver ces résultats de tests ci-dessous ;

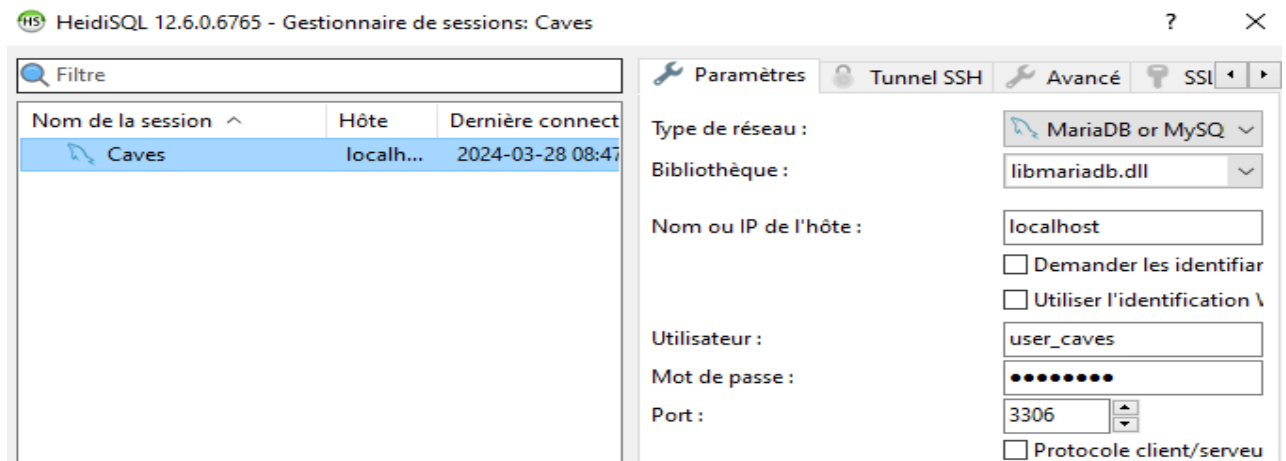
TEST		
Travaux à réaliser	Réalise	Parties manquantes
Avertissement que l'utilisateur doit renseigner les informations lors du processus d'inscription.	Un avertissement apparaîtra à l'écran : Ne le laissez pas vide.	
Plusieurs personnes peuvent-elles s'inscrire avec la même adresse e-mail ?	Si le même e-mail a déjà été utilisé, il affichera un avertissement utilisateur existe.	
Si le champ non obligatoire n'est pas renseigné, y aura-t-il une erreur dans la base de données ?	Si les champs qui peuvent être laissés vides ne sont pas remplis, ces données dans la base de données resteront vides.	
Avertissement si les informations saisies par l'utilisateur ne correspondent pas au type de données.	Si les informations appropriées ne sont pas saisies pour les types de données tels que les e-mails ou les numéros, un avertissement de type de données incorrect apparaît à l'écran.	
Les informations saisies lors de l'inscription ont-elles été placées dans les colonnes appropriées de la base de données ?	Les informations sont au bon endroit et la fonction Auto Incrimination est active.	
Le mot de passe de l'utilisateur est-il haché et enregistré dans la base de données ?	Le mot de passe a été enregistré sous forme hachée.	
Est-ce que cela donne un avertissement si le type d'e-mail n'est pas valide pendant le processus de connexion ?	Lors de la connexion par e-mail, un avertissement apparaît pour un type d'e-mail inapproprié.	
Y a-t-il un avertissement si le mot de passe saisi lors du processus de connexion ne correspond pas au mot de passe saisi au moment de l'inscription ?	L'utilisateur est averti de saisir le mot de passe correct et ne peut pas se connecter à moins que le mot de passe et les informations de courrier électronique corrects correspondent à la base de données.	
Une fois le processus de connexion terminé, y a-t-il un avertissement de connexion réussie à l'écran ?	L'utilisateur est informé par un écran d'avertissement qu'il s'est connecté.	
Les boutons changent-ils après le processus de connexion ?	Les boutons changent et de nouvelles fonctions apparaissent après le processus de connexion.	
Le bouton filtre est-il fonctionnel ?	Lorsque vous cliquez sur le bouton de filtre, une fenêtre contextuelle de filtre publicitaire avec le titre du titre apparaît, par exemple. Cette pop-up répertorie les annonces en fonction de leur titre et les affiche sur l'écran.	Pendant le processus de filtrage, d'autres publicités que le titre filtré sont affichées sur l'écran de la console ; de nouvelles études et tests sont nécessaires à ce stade.

## 3 Implémentation

### 3.1 Choix techniques

- Frontend
- Backend
- Système de base de données

La base de données MySQL a été utilisée pour le projet.



Tableaux : La base de données est conçue à partir de divers tableaux tels que les utilisateurs, les publicités, les produits, les adresses et les photos. Le modèle relationnel a été utilisé pour maintenir l'intégrité des données et assurer leur cohérence.

caves		240.0 KiB
★	announcements	48.0 KiB
	categories	32.0 KiB
	comments	32.0 KiB
	messages	64.0 KiB
	photos	32.0 KiB
	products	16.0 KiB
	users	16.0 KiB

Nom	Lignes	Taille
announcements	8	48.0 KiB
categories	6	32.0 KiB
comments	0	32.0 KiB
messages	0	64.0 KiB
photos	8	32.0 KiB
products	8	16.0 KiB
users	2	16.0 KiB

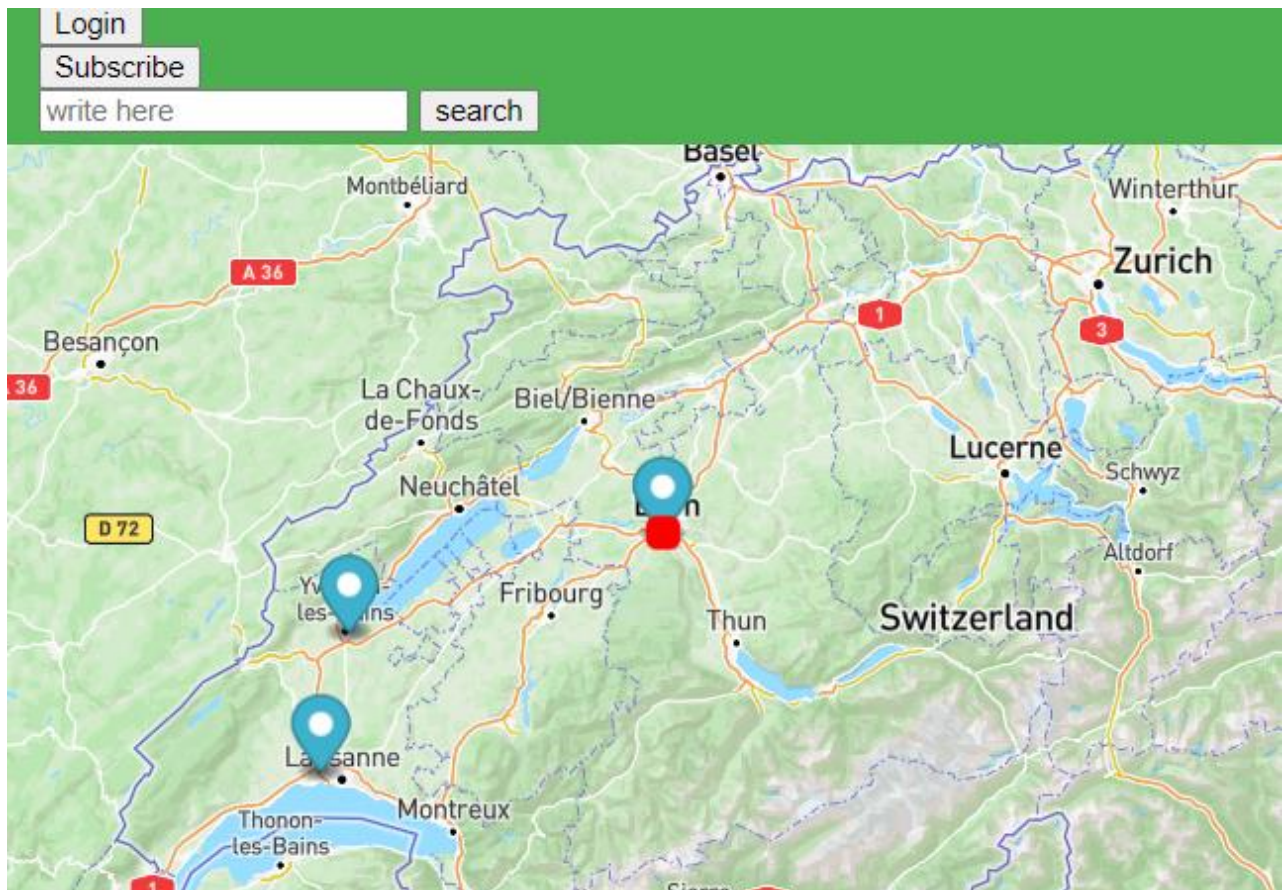


#	Nom	Type de données	Taille/Ensem...	Non signé	NULL autorisé	ZEROFILL	Par défaut
1	id	INT	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT
2	buyer_id	INT	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Pas de défaut
3	seller_id	INT	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Pas de défaut
4	announcement_id	INT	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Pas de défaut
5	message_text	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Pas de défaut
6	sent_time	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	'CURRENT_TIMESTAMP'

## - Map

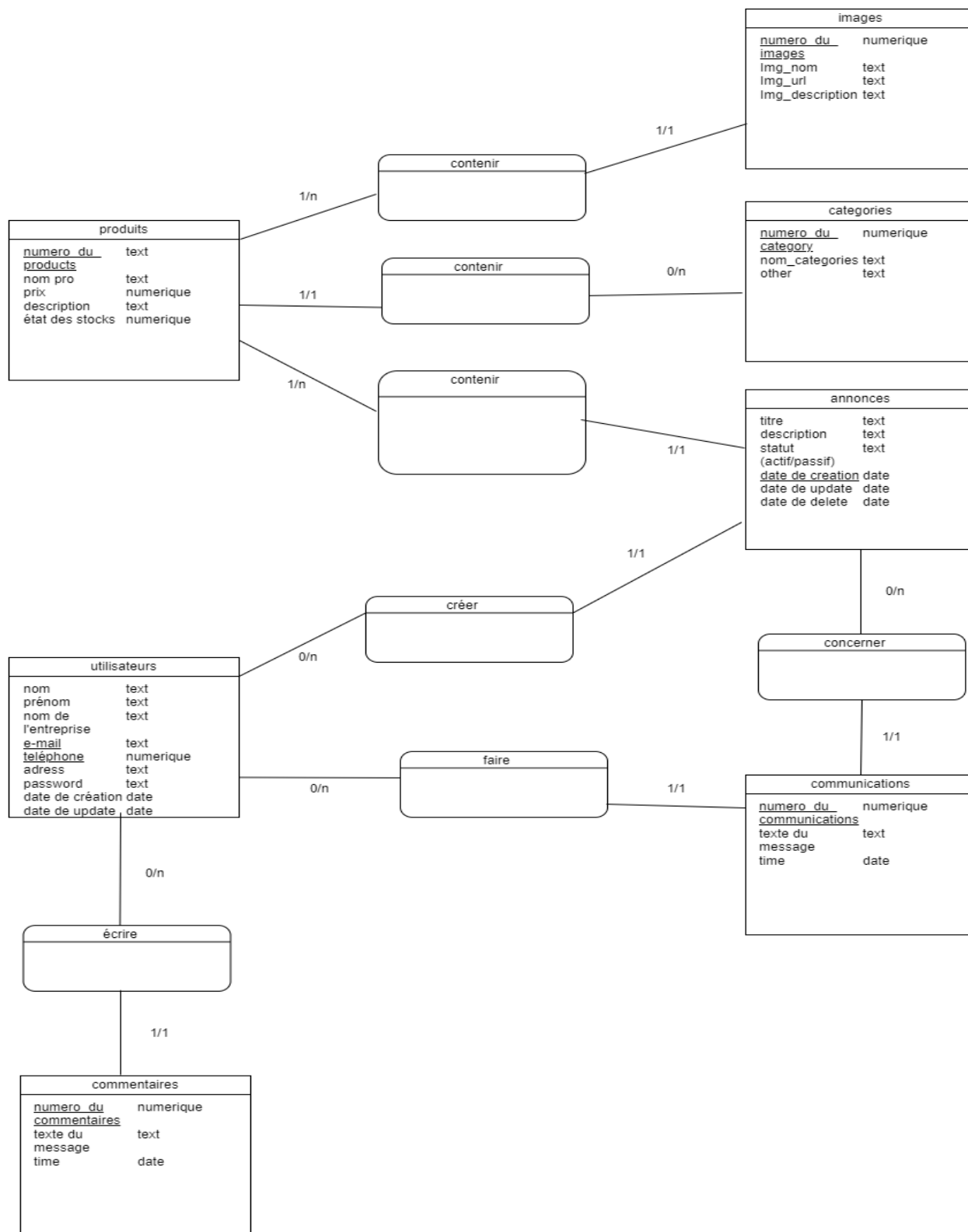
Mapbox GL : utilisé comme composant cartographique. Mapbox GL est un outil puissant pour créer des cartes interactives basées sur le Web. Nous l'utilisons pour visualiser les listes des utilisateurs ainsi que leur emplacement.

Domaines d'utilisation : scénarios tels que les utilisateurs visualisant leurs publicités sur la carte, trouvant des publicités à proximité et effectuant des recherches basées sur la localisation.



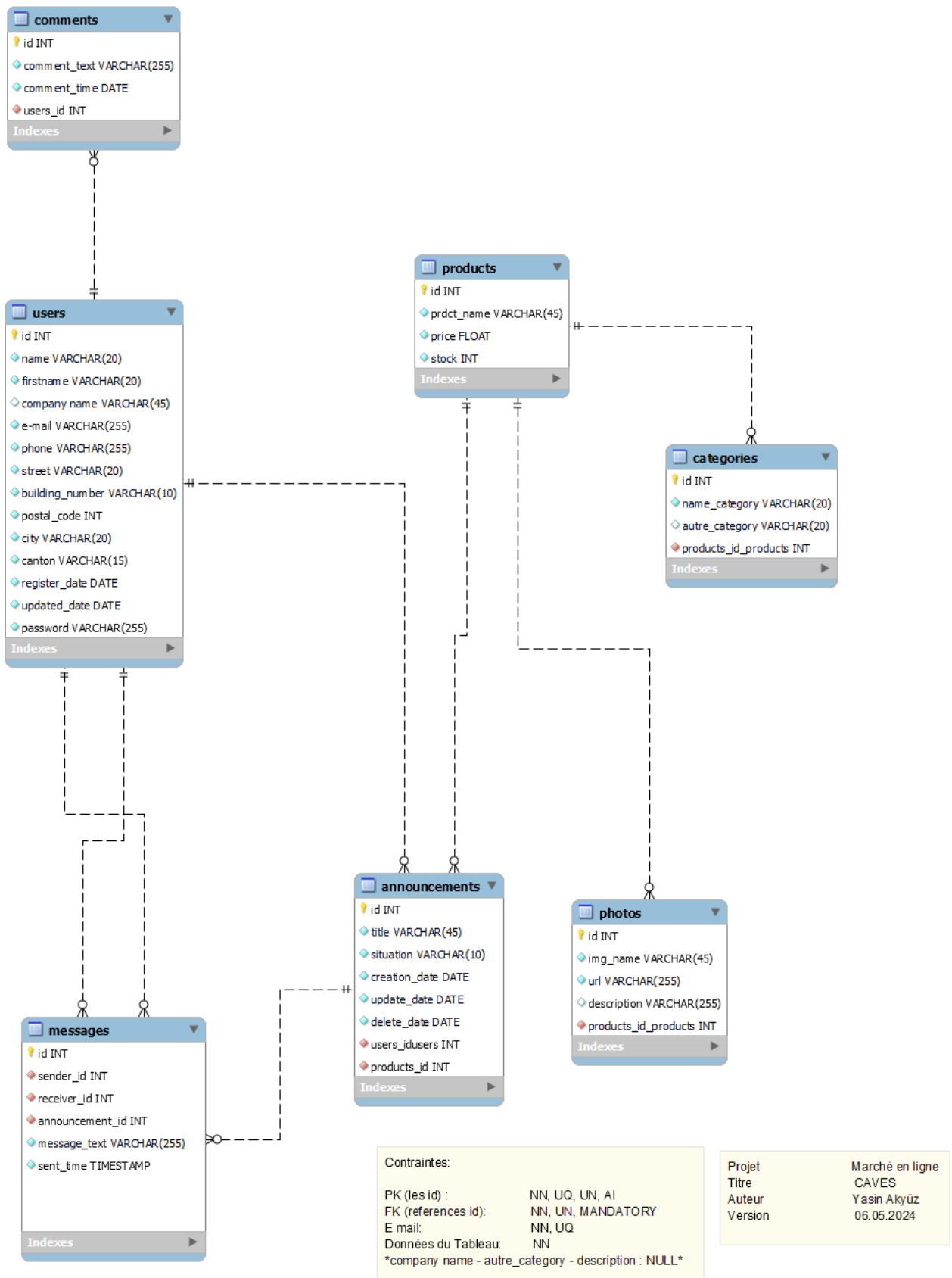
- Filtre
- Search
- Message

## 3.2 Modèle conceptuel de données



Projet	Marché en ligne
Titre	CAVES
Auteur	Yasin Akyüz
Version	06.05.2024 V1.0

### 3.3 Modèle Logique de données



Le schéma de la base de données est constitué de tables qui définissent diverses entités et les relations entre elles. Les entités de la base de données sont listées comme suit : produits, users,

comments, photos, catégories, annonces, et messages. Chaque table est équipée de différentes contraintes nécessaires pour assurer l'intégrité des données.

#### - Tables et Relations :

Chaque table possède une clé primaire (id) définie avec des contraintes telles que NN (Not Null), UQ (Unique), UN (Universal), AI (Auto Incrément), ce qui assure que chaque enregistrement de la table soit identifié par une clé unique et auto-incrémentée.

Les clés étrangères (FK) se basent sur les id référencés et comprennent des contraintes NN (Not Null), UN (Universal), MANDATORY, indiquant que les références aux tables liées sont obligatoires et que chaque référence doit être unique.

Les adresses e-mail sont définies comme NN (Not Null) et UQ (Unique), signifiant que chaque utilisateur doit avoir une adresse e-mail unique.

Il est indiqué que certains champs peuvent être NULL, ce qui montre que ces zones sont optionnelles.

```
CREATE TABLE IF NOT EXISTS `mydb`.`messages` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `buyer_id` INT UNSIGNED NOT NULL,
  `seller_id` INT UNSIGNED NOT NULL,
  `announcement_id` INT UNSIGNED NOT NULL,
  `message_text` VARCHAR(255) NOT NULL,
  `sent_time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`),
  INDEX `fk_messages_sender_id_idx` (`buyer_id` ASC) VISIBLE,
  INDEX `fk_messages_receiver_id_idx` (`seller_id` ASC) VISIBLE,
  INDEX `fk_messages_announcement_id_idx` (`announcement_id` ASC) VISIBLE,
  CONSTRAINT `fk_messages_sender`
    FOREIGN KEY (`buyer_id`)
      REFERENCES `mydb`.`users` (`id`)
      ON DELETE CASCADE
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_messages_receiver`
    FOREIGN KEY (`seller_id`)
      REFERENCES `mydb`.`users` (`id`)
      ON DELETE CASCADE
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_messages_announcement`
    FOREIGN KEY (`announcement_id`)
      REFERENCES `mydb`.`announcements` (`id`)
      ON DELETE CASCADE
      ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

#### - Normalisation et Critères BCNF :

Les tables conformes à la 1NF montrent une structure où les valeurs des colonnes sont atomiques et où il n'y a pas de groupes répétitifs.

Les conditions 2NF et 3NF sont supposées être remplies en considérant que tous les champs, à l'exception des clés étrangères, dépendent de la clé de la table.

La BCNF exige que tous les déterminants soient une partie de la clé candidate. Selon les informations indiquées dans le schéma, chaque colonne id sert de clé primaire pour la table et les clés étrangères se réfèrent correctement aux clés des tables concernées.

users - Table										
Table Name: users										
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
firstname	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
company name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
e-mail	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
phone	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
street	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
building_number	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
postal_code	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
city	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
canton	VARCHAR(15)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
register_date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
updated_date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
password	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

annoncements - Table										
Table Name: annonces										
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
title	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
situation	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
creation_date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
update_date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
delete_date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
users_idusers	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
products_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Nous devons spécifier des contraintes lors de la saisie des données dans le tableau MLD. Vous pouvez voir des exemples de contraintes sélectionnées à la fois dans la description en bas du tableau mld et dans le tableau des utilisateurs et annonces donné en exemple.

### 3.4 Diagramme de Flux

Ce diagramme de flux illustre comment les utilisateurs peuvent effectuer diverses opérations sur la plateforme, étape par étape. Il est composé de processus qui définissent les actions des utilisateurs et comment le système y répond.

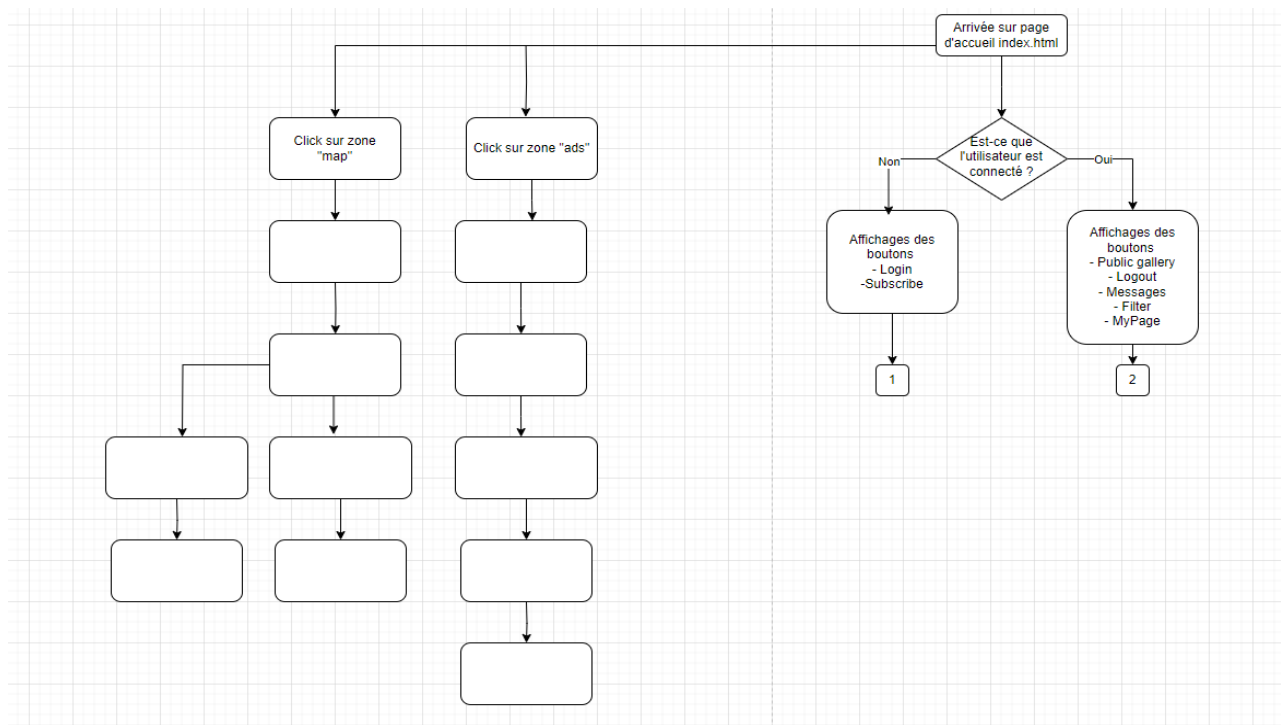
Pour décrire les processus affichés dans le diagramme par exemple :

- L'utilisateur entre sur le site et peut voir divers boutons : connexion (login), inscription, etc.
- L'utilisateur démarre le processus de connexion en cliquant sur le bouton "Login".
- Si les informations de l'utilisateur sont correctes, le système reconnaît l'utilisateur et ouvre la

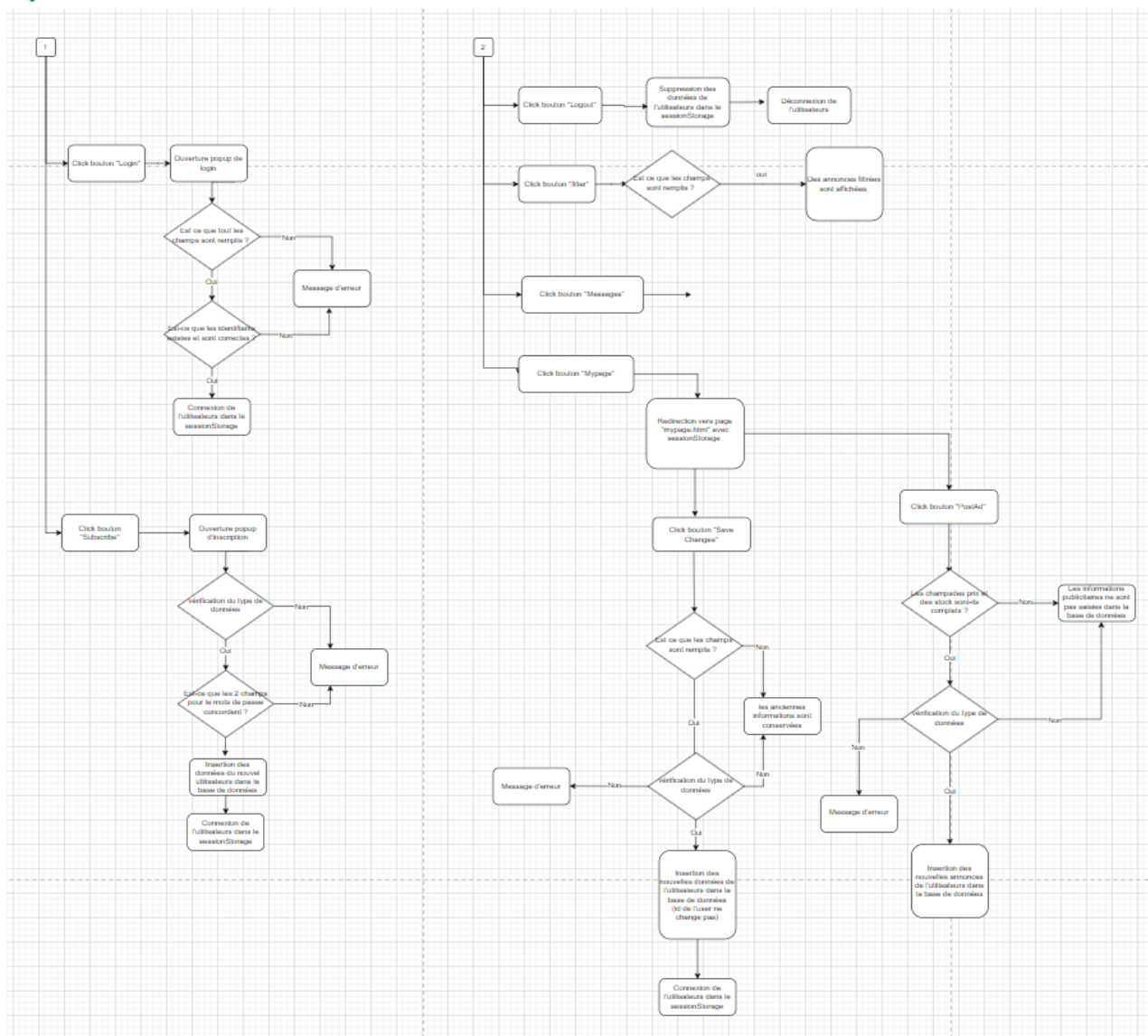
session. Sinon, il répond avec un message d'erreur.

- Une fois la session ouverte, diverses options sont proposées à l'utilisateur : mise à jour des informations personnelles, envoi de messages, déconnexion, etc.
- L'utilisateur peut fermer la session en cliquant sur le bouton "Log out".

Les actions de l'utilisateur et les réponses du système dépendent de certaines conditions, telles que la véracité du nom d'utilisateur et du mot de passe, le remplissage des champs du formulaire et l'exactitude des informations dans la base de données. Chaque étape dirige à la suivante en fonction des conditions positives ou négatives.







### 3.5 Points techniques spécifiques

#### 3.5.1 Enregistre les images

La manière dont les photos sont stockées et gérées sur les sites Web dépend souvent des besoins et des exigences de performances de l'application. Dans votre cas, vous envisagez de stocker des photos sous forme de texte dans la base de données plutôt que de les stocker sous forme de fichiers physiques. Les deux méthodes présentent des avantages et des inconvénients :

Stockage sur le système de fichiers :

Avantages :

- Le système de fichiers est généralement optimisé pour stocker des fichiers multimédias volumineux.
- Le serveur Web peut servir directement des fichiers statiques, ce qui est souvent plus rapide que d'extraire des données de la base de données.
- Il permet de déplacer facilement les images vers des services tels que CDN (Content Delivery



Network).

Désavantages :

- S'il existe de nombreux fichiers, le système de fichiers peut devenir difficile à gérer.
- Les noms et chemins de fichiers doivent être conservés dans la base de données, ce qui peut entraîner des problèmes de synchronisation.
- Les opérations de sauvegarde et de migration peuvent devenir plus complexes.

Stockage dans la base de données (par exemple sous forme de texte BLOB ou encodé en base64) :

Avantages :

- Les métadonnées et le contenu des fichiers sont conservés au même endroit, ce qui peut faciliter la gestion.
- La sauvegarde et la migration peuvent être plus simples car toutes les données sont consolidées dans la base de données.

Désavantages :

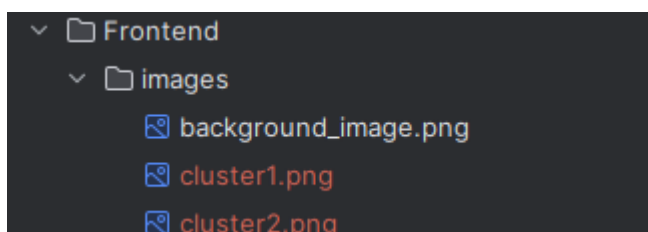
- Les bases de données ne sont généralement pas optimisées pour stocker de grandes quantités de données binaires.
- Vous devez écrire du code supplémentaire pour servir le contenu du fichier, car le serveur Web ne peut pas le servir directement.
- La charge sur la base de données augmente, ce qui peut entraîner des problèmes de performances et d'évolutivité.

L'idée de stocker du texte dans la base de données est généralement privilégiée pour les petits fichiers ou un petit nombre de fichiers multimédias. Mais dans un système à grande échelle, il est plus courant de conserver physiquement les fichiers et leurs références dans la base de données.

Hôte: localhost Base de données: caves Table: photos Données Requête\*

caves.photos: 24 ligne(s) au total (exact)

#	id	img_name	url	description	products_id_products
21	35	cluster1.png	C:\ysa\Trimestre-3\Pré-tpi\Frontend\images\cluster2.png	Your description here	51
22	36	cluster2.png	C:\ysa\Trimestre-3\Pré-tpi\Frontend\images\cluster2.png	Your description here	51
23	37	cluster1.png	C:\ysa\Trimestre-3\Pré-tpi\Frontend\images\cluster1.png	Your description here	52
24	38	cluster2.png	C:\ysa\Trimestre-3\Pré-tpi\Frontend\images\cluster2.png	Your description here	52



- Comment traiter le dossier sur internet

```
$finfo=newfinfo(FILEINFO_MIME_TYPE);
$fileMimeType = $finfo->file($_FILES['url']['tmp_name'][$i]);
```

```
$finfo = new finfo(FILEINFO_MIME_TYPE); :
```

Crée une instance de la classe finfo et définit cette instance en mode de détermination de type MIME. La classe finfo est utilisée pour vérifier le type MIME des fichiers.

```
$fileMimeType = $finfo->file($_FILES['url']['tmp_name'][$i]); :
```

En utilisant le nom temporaire du fichier téléchargé, détermine le type MIME de ce fichier. Le chemin `$_FILES['url']['tmp_name'][$i]` indique le chemin où le fichier téléchargé via le formulaire est temporairement stocké sur le serveur.

- L'extension PHP finfo doit être activée en PHP.

->php.ini

```
912 ;extension=ftp
913 extension=fileinfo
914 ;extension=gd2
```

### 3.5.2 Carte

### 3.6 Bugs

## 4 Conclusions

### 4.1 Objectifs atteints

### 4.2 Objectifs non-atteints

### 4.3 Suites possibles au pour le projet

## 5 Annexes

### 5.1 Journal de travail

## 6 SOURCES

<http://bilgisayar-muhendisleri.blogspot.com/2014/01/php-mysql-image-upload-etme-ve-okuma.html>  
<https://kodusta.com/blog/php-pdo-ckeditor-resim-yukleme-ve-veritabanina-kaydetme>

[https://www.w3schools.com/php/php\\_mysql\\_insert\\_multiple.asp](https://www.w3schools.com/php/php_mysql_insert_multiple.asp)

<https://stackoverflow.com/questions/68038388/how-to-upload-image-on-mysql-database-using-php>

<https://www.burakalpikara.com/mapbox-ile-projemize-harita-eklemek>

[https://mesutd.com/php-mysql-baglanma-mysqli-ekleme-silme-duzenleme-listeleme-dosya-yukleme#google\\_vignette](https://mesutd.com/php-mysql-baglanma-mysqli-ekleme-silme-duzenleme-listeleme-dosya-yukleme#google_vignette)

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/JSON/stringify](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify)

<https://docs.mapbox.com/mapbox-gl-js/guides/>