

DATE

---

# LUCH VINTAGE

SHOP EN LIGNE DE  
VETEMENTS  
SECONDE MAIN

---

[WWW.LUCH-VINTAGE.FR](http://WWW.LUCH-VINTAGE.FR)







# Sommaire

03 Présentation

06 Introduction

08 Background

14 Conception et design

27 Présentation  
fonctionnelle

36 Développement

44 Demonstration

49 Conclusion



# Présentation

# Rascle Romain

## Mon parcours

---

J'ai suivi des études supérieures dans la restauration et construit mon parcours professionnel dans ce domaine jusqu'à occuper un poste de responsable. Avec le temps, j'ai ressenti le besoin de changement et me suis orienté vers la vente, mais cela ne correspondait pas non plus à mes attentes.

J'ai alors pris du recul pour me recentrer sur ce qui me passionne vraiment, sur mes valeurs et sur la manière dont je souhaite m'épanouir professionnellement.

---

## Mon objectif

Je veux retrouver le plaisir de créer quelque chose dans un univers qui me passionne. L'idée d'être constamment challengé par l'évolution et les défis me motive particulièrement.





# Notre équipe



**Rascle Romain**  
DEVELOPPEUR



**Lucie Chabran**  
ADMINISTRATRICE

# Introduction



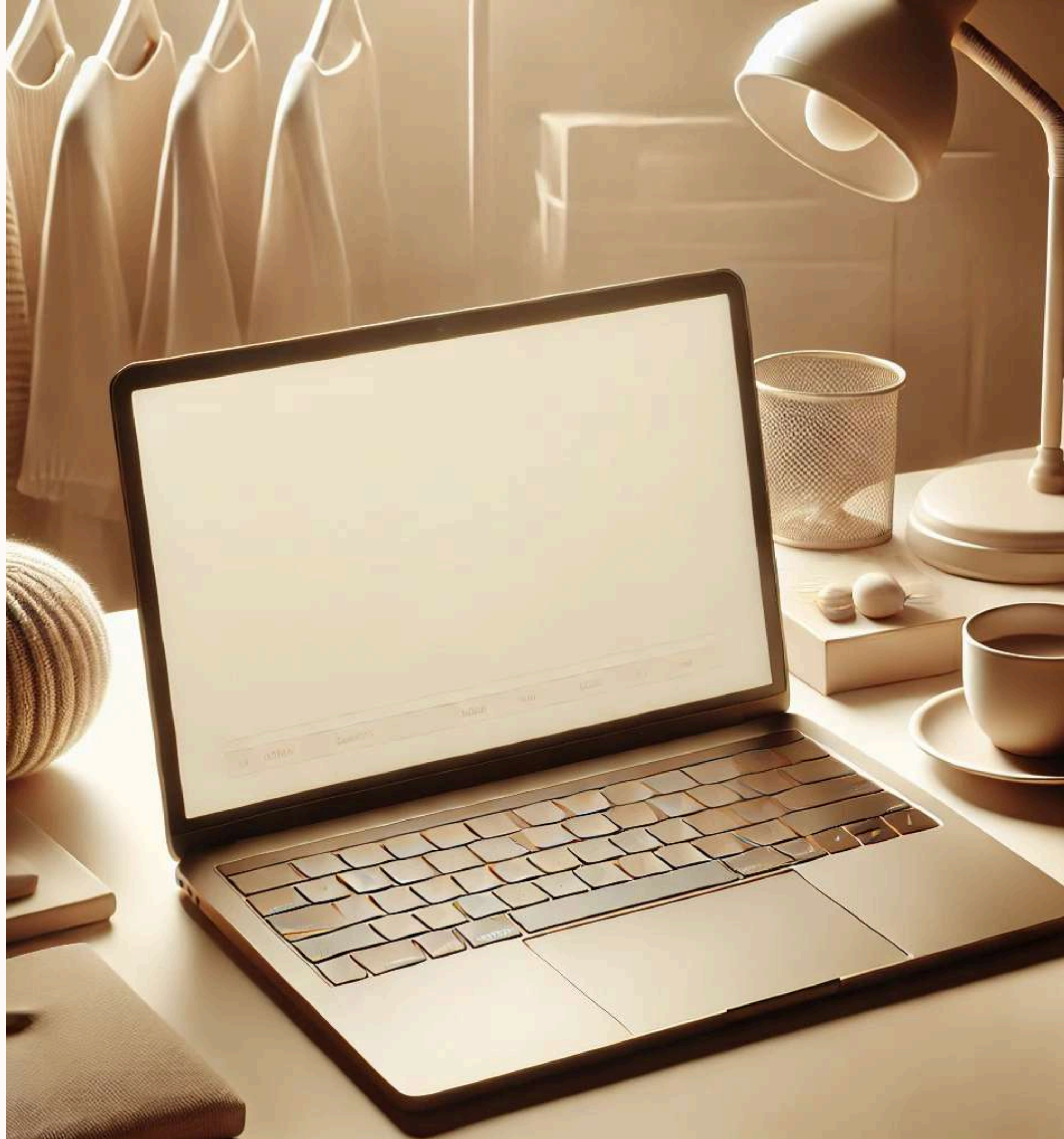
# Le projet

Luch Vintage, sera une boutique en ligne dédiée à la mode de seconde main. Elle propose une sélection unique de vêtements et accessoires vintage, soigneusement choisis pour leur style rétro et leur qualité. Luch Vintage permet à ses clients de faire des économies, d'entretenir leurs vêtements et faire un geste pour la planète tout en enrichissant leur garde-robe avec des pièces authentiques et durables.

## La problématique

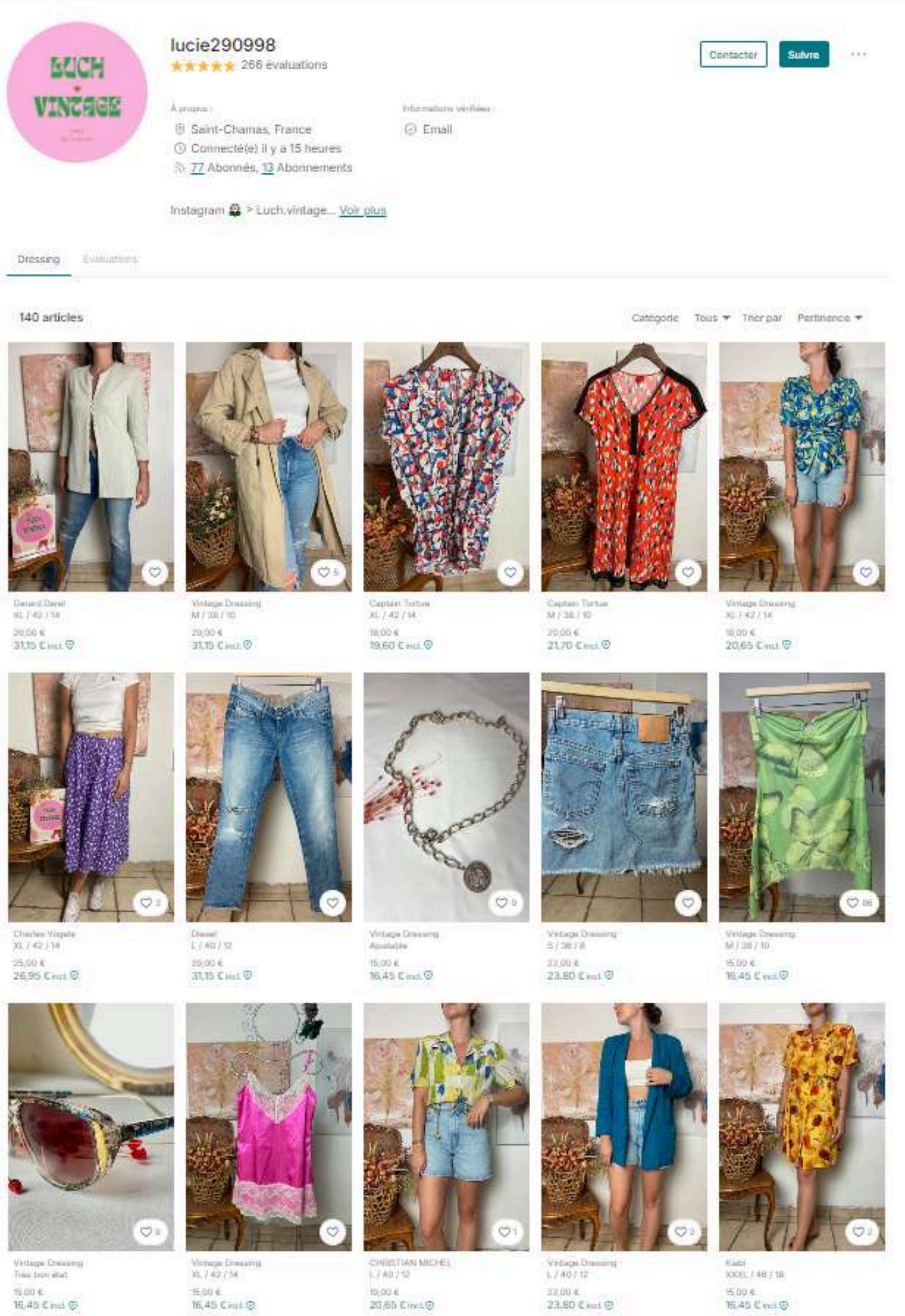
Luch.Vintage est déjà en activité sur vinted néanmoins nous faisons face à plusieurs problématiques :

- Concurrence forte
- Frais de port très élevés
- Bloqué par la charte visuelle de Vinted (identité visuelle faible)
- Cycle de vente dépendant de Vinted (aucune autonomie)
- Captation via les réseaux faibles



# Background






# Vinted

<https://www.vinted.fr/member/16605650>

- Identité faible
- Impossible de changer de nom
- Mise en avant et organisation payante





28,00 €  
**30,10 €**  
Inclut la Protection acheteurs

Jusqu'à -28 % sur les frais de port

MARQUE	RALPH LAUREN	⋮
TAILLE	L / 40 / 12	⊗
ÉTAT	TRÈS BON ÉTAT	⊗
COULEUR	MULTICOLORE	
LOCALISATION	SAINT-CHAMAS, FRANCE	
OPTIONS DE PAIEMENT	CARTE BANCAIRE	
NOMBRE DE VUES	25	
AJOUTÉ	IL Y A 4 MOIS	

**Blouse Ralph Lauren**  
Blouse 🌈

- . Ralph Lauren
- . Coupe cache coeur
- . Multicolore
- . Matière légère
- . Se ferme avec un noeud

🚀 envoi rapide  
Disponible pour plus d'informations

Envoi à partir de 1,95 €

Profite de réductions jusqu'à -28% sur la livraison en point relais. Plus d'informations lors

+ 3

3

Accueil / Femmes / Vêtements / Hauts et t-shirts / Blouses / Ralph Lauren Blouses

(140) articles disponibles

Acheter un lot  
Jusqu'à 15% de réduction

Voir tous les articles

Créer un lot

## Vends ton article

Ajoute jusqu'à 20 photos. [Voir astuces.](#)

+ Ajoute des photos

Titre

ex : Chemise Sézane verte

Décris ton article

ex : porté quelques fois, taille correctement

Catégorie

Sélectionne une catégorie

Prix

0,00 €

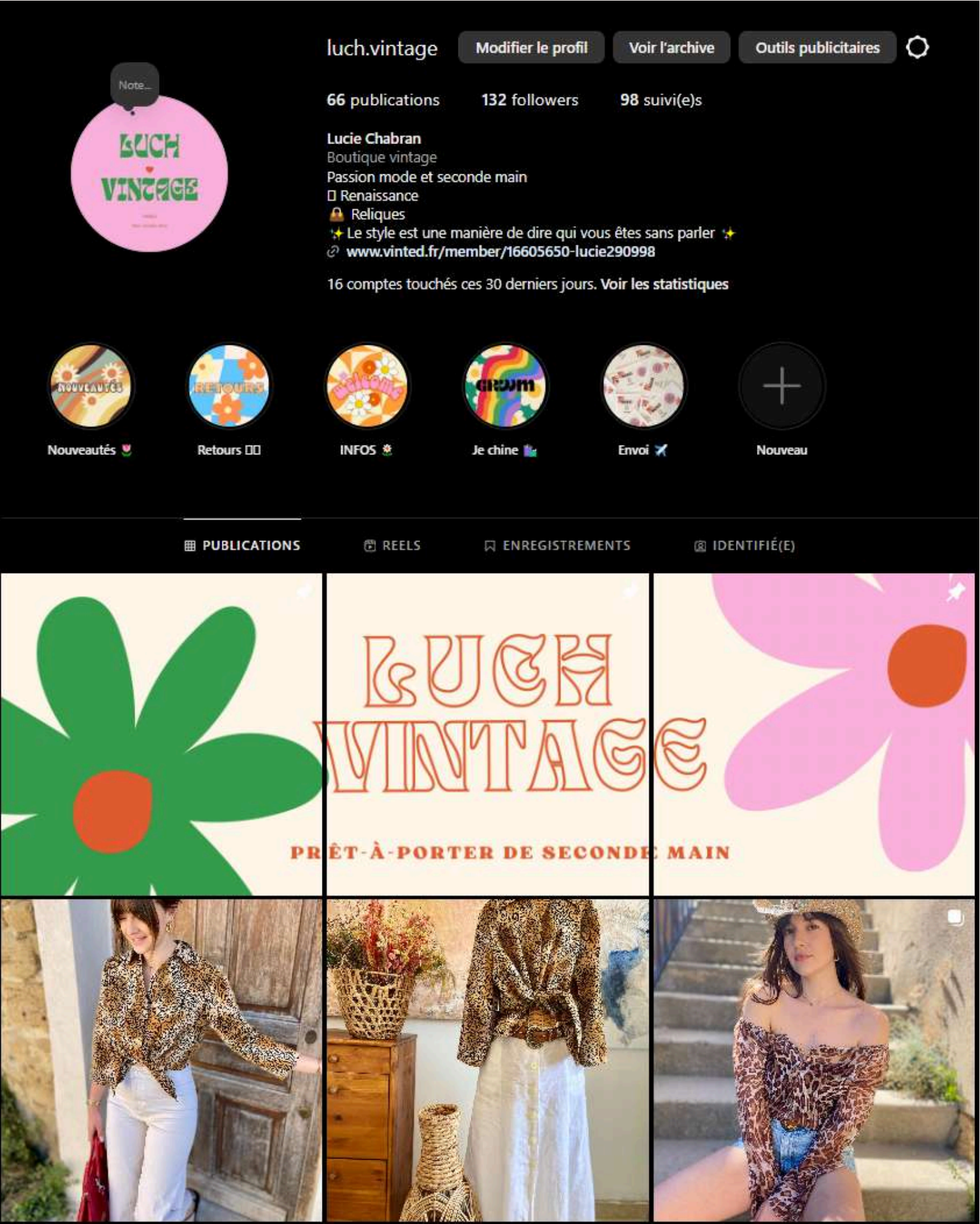
Que penses-tu de notre procédure pour ajouter de nouveaux articles ?

Donner un avis

<https://www.vinted.fr/member/16605650>

- Présentation impersonnelle
- Système de post des photos et description rigide





# INSTAGRAM

- Identité forte
- Captage de communauté
- Mise en avant ‘pro’
- Ordre des posts pour créer des collections





# INSTAGRAM

- Ordre des posts pour créer des collections
- Mise en avant d'un style





# Les objectifs

## DÉVELOPPEMENT

Développer une solution simple et fluide avec une identité simple, compréhensible et clair

## UTILISATION

Créer un outils simple et efficace pour que l'administrateur puisse gerer réseaux/site seul

## DURABILITÉ

Créer une boucle entre la vie de la seconde main ( guide, astuces, story, post ) sur les réseau et la transverser vers le site pour mener à l'achat en créant un chemin fermée pour l'utilisateur



# Conception et design



# Les fonctionnalités

- Gestion des utilisateurs
- Gestion des produits
- Gestion des commandes
- Système de paiement
- Livraison

- Gestion des retours
- Gestion des promotions
- Système de notification
- Administration



# User Experience



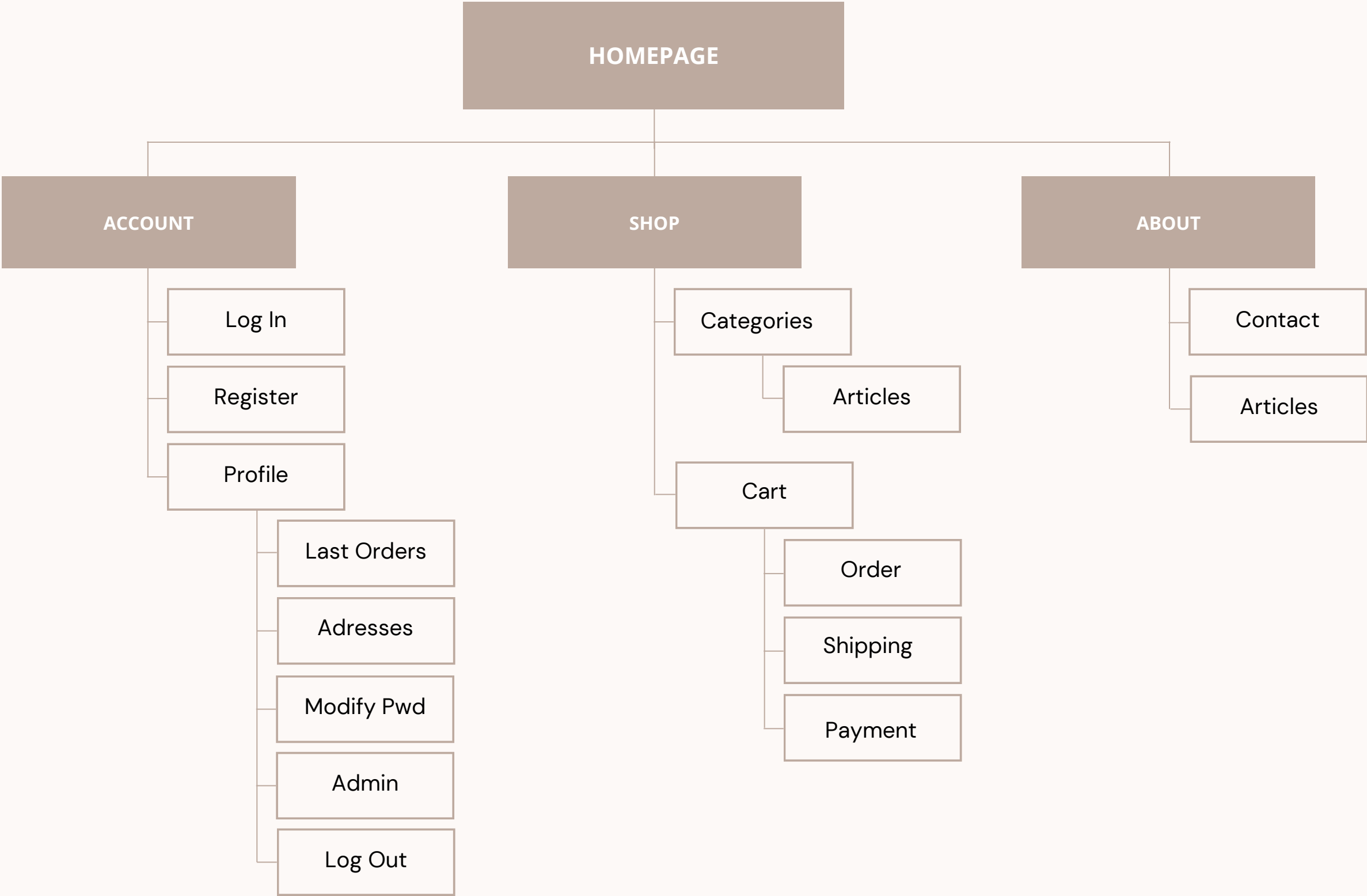
# Outils

- JMerise
  - Figma
-



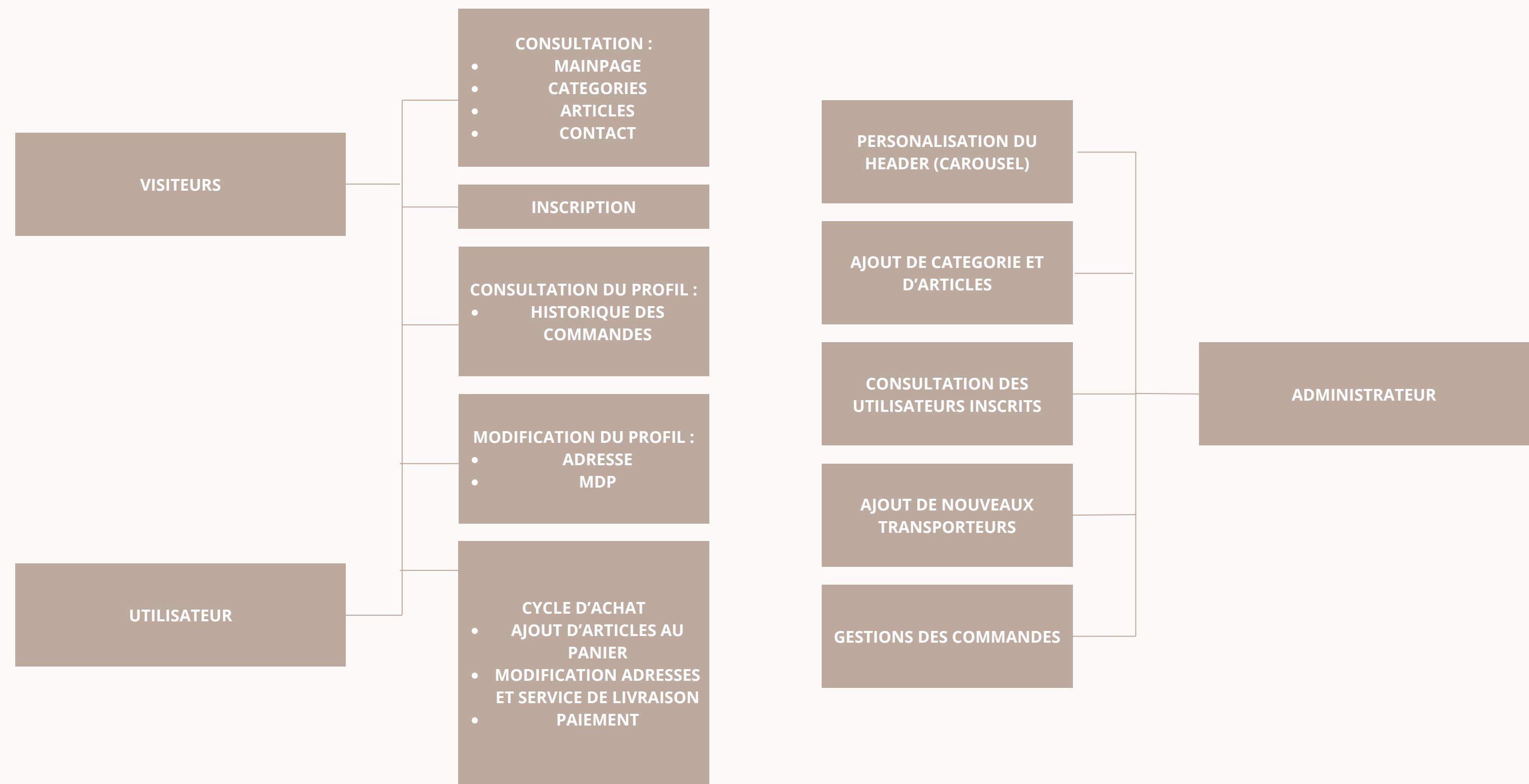
# Arborescence

UML : Unified model Language ( Language de  
Modélisation unifié ) tous les diagrammes font partie  
de cette classification





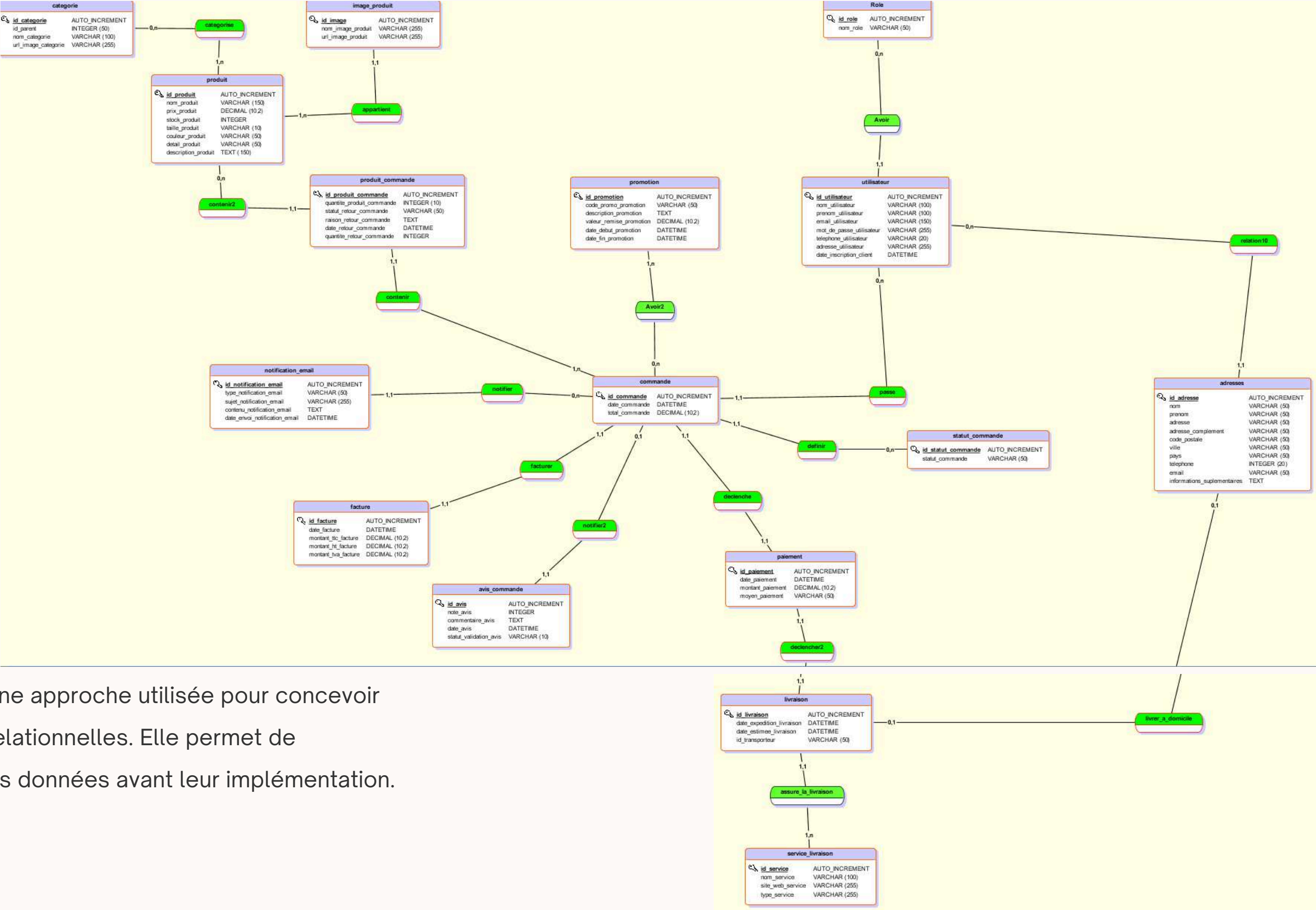
# Diagramme Utilisateurs





# MCD

Modele conceptuelle de  
donnée



La méthode Merise est une approche utilisée pour concevoir  
des bases de données relationnelles. Elle permet de  
structurer et organiser les données avant leur implémentation.





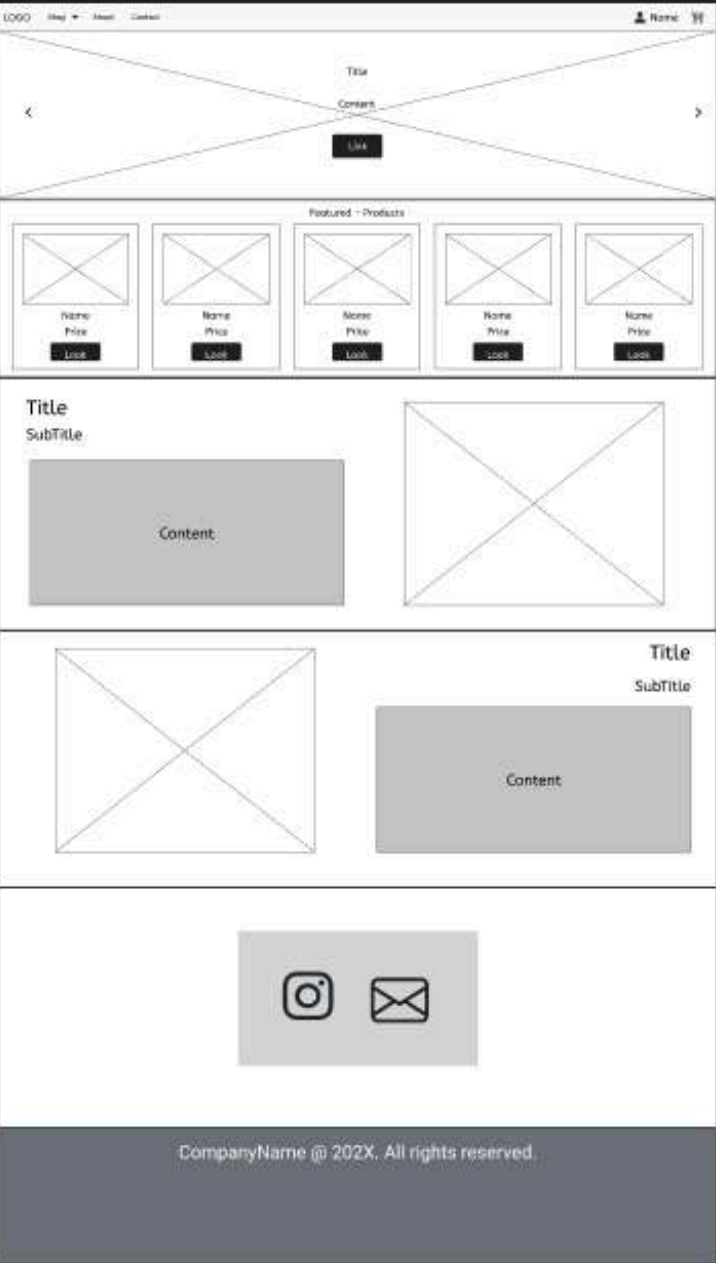
# Wireframe Mobile

On travail sur la logique mobile first donc on créer  
avant le wireframe mobile puis on s’adapte a des  
formats plus grands

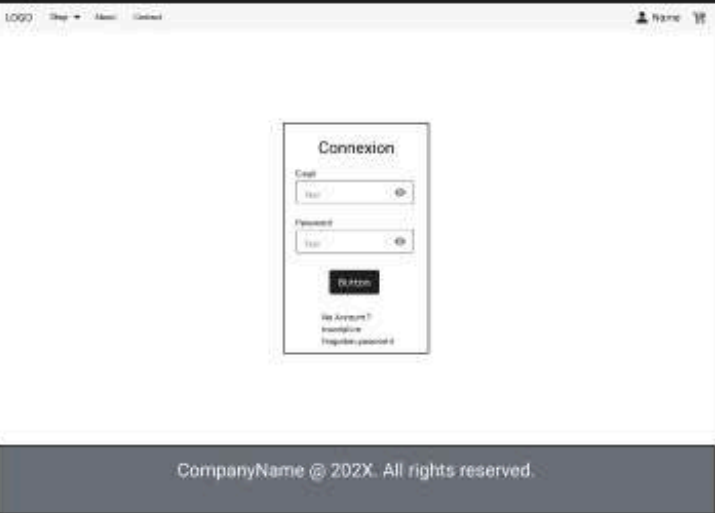


# Wireframe Desktop

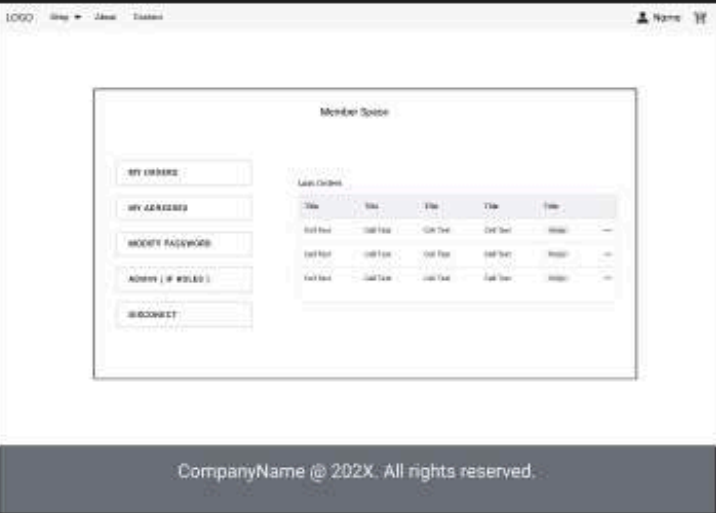
Main Page



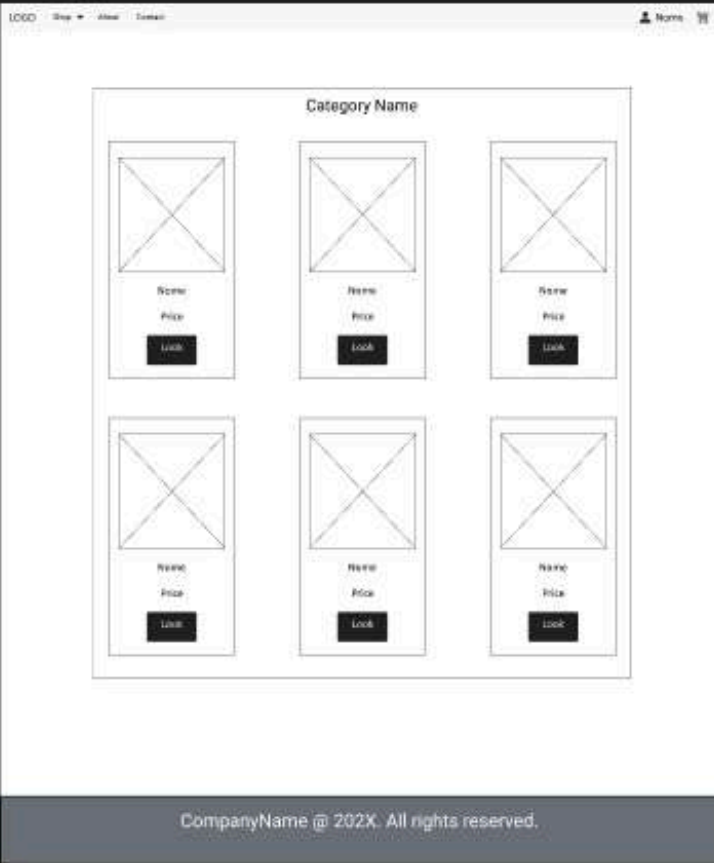
Login



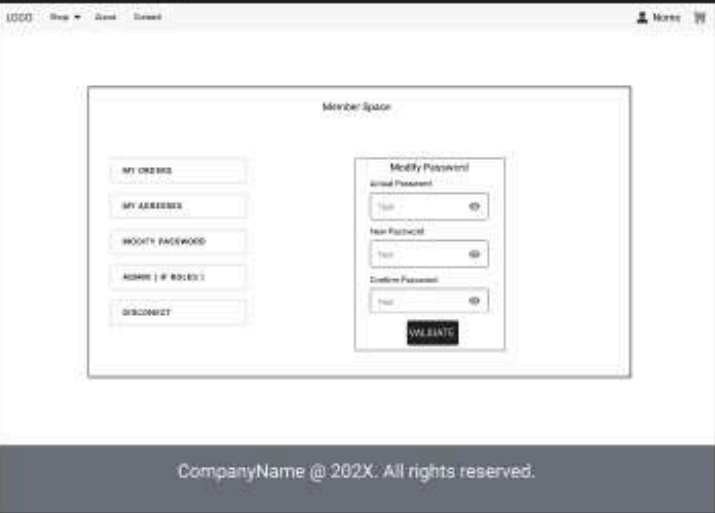
Account - Main



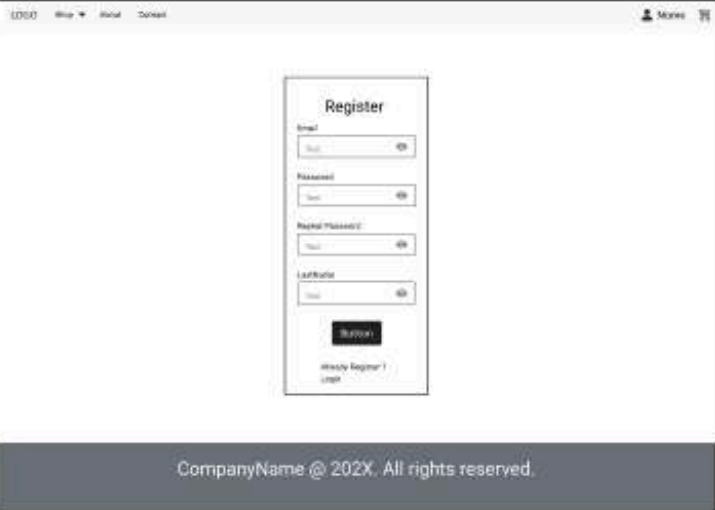
Category



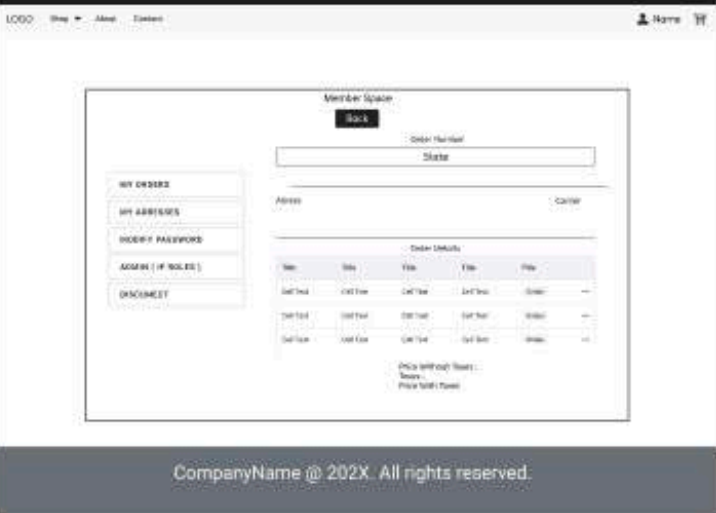
Account - Modify Password



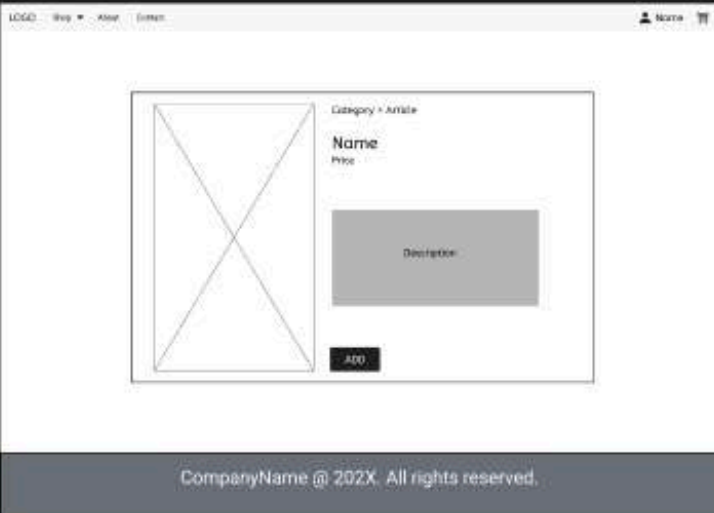
Register



Account - Orders Details



Article



Forgotten Password



Account - Addresses

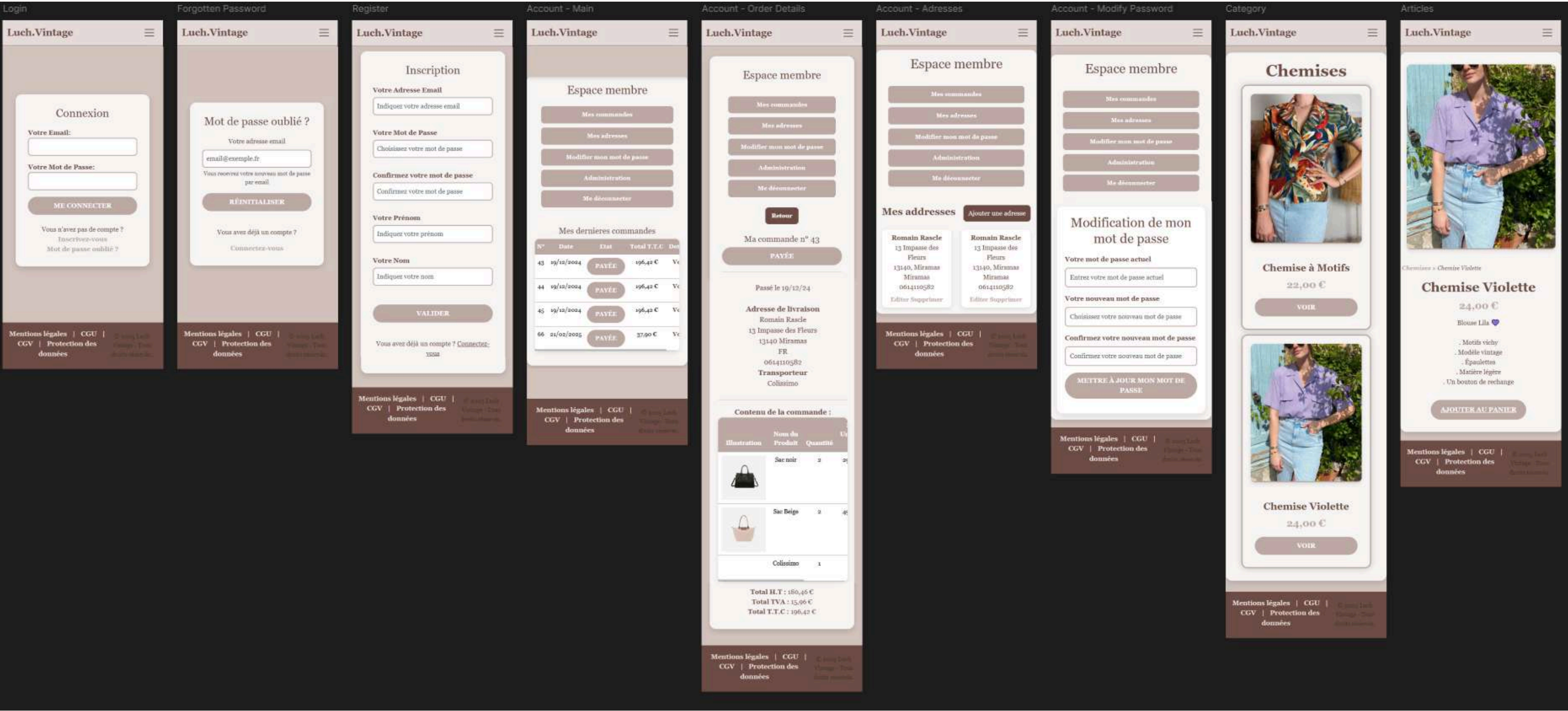




# Moodboard

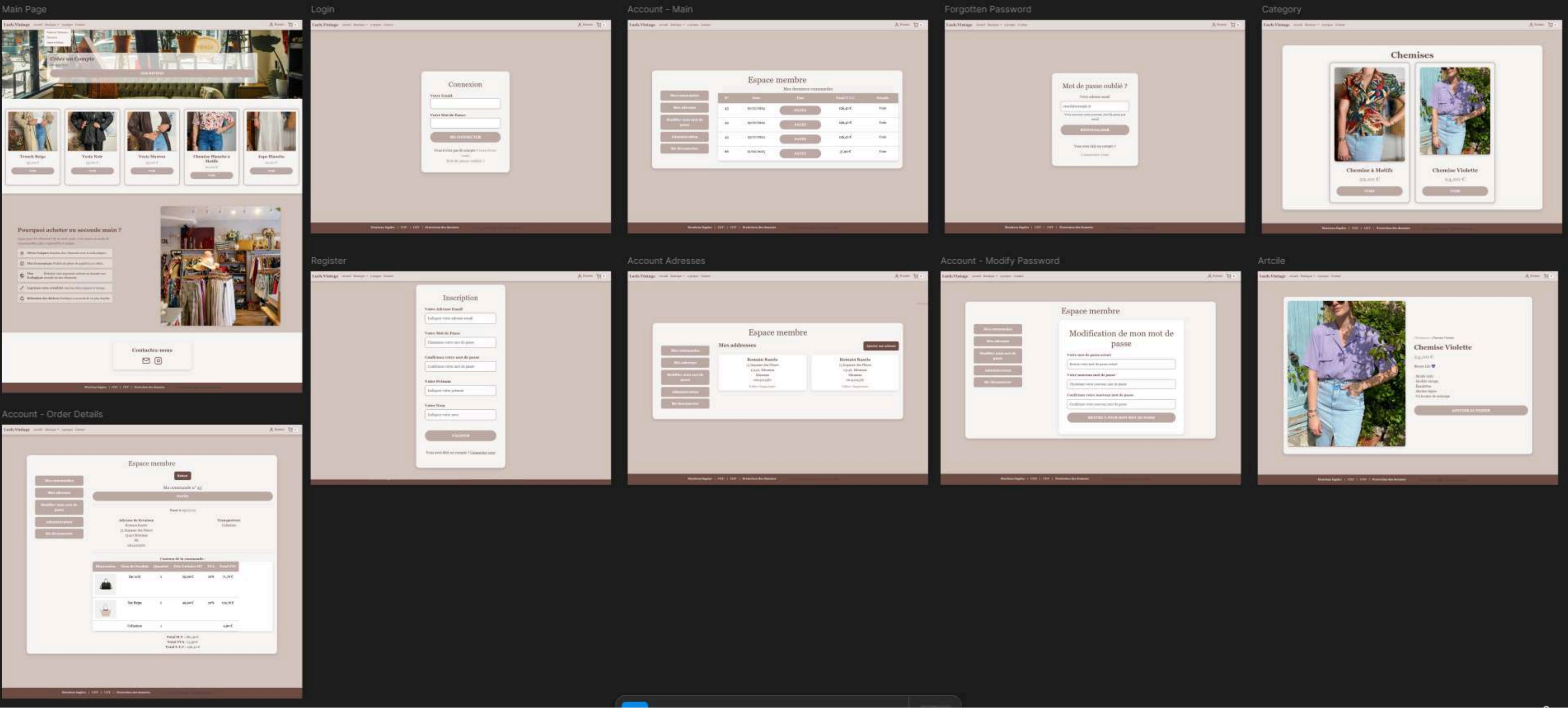


# Maquette Mobile





# Maquette Desktop



# Presentation fonctionnelle ( UX / MCD )



# Production

# Outils et Methodes



# Methode

- Méthode agile ( Sprint )

# Outils

- Trello
- Github
- Hostinger

# Methode agile / Trello

Sprint 1 : Initialisation du projet

DONE

Mise en place du cahier des charges

DONE

Mettre en place Trello avec la structure des sprints

DONE

Créer un repository Git/GitHub pour le projet

DONE

Initialiser le projet Symfony

DONE

Configurer l'environnement de développement local (installation des dépendances, serveurs, etc.)

+ Ajouter une carte

Sprint 2 : Gestion des utilisateurs et sécurité

DONE

Installer et configurer Symfony Security

DONE

Créer l'entité User avec les champs nécessaires (nom, prénom, email, etc.)

DONE

Configurer le système d'inscription et de connexion

DONE

Mettre en place la gestion des rôles (client, admin)

DONE

Tester les fonctionnalités d'authentification

+ Ajouter une carte

Sprint 3 : Gestion des produits et Categories

DONE

Créer les entités Produits et Catégories

DONE

Créer un CRUD pour la gestion des produits

DONE

Créer un CRUD pour la gestion des catégories

DONE

Ajouter la fonctionnalité d'association des images aux produits

DONE

Implémenter les relations entre produits et catégories

+ Ajouter une carte

Sprint 4 : Gestion du panier, commandes et paiements

DONE

Créer les entités Panier et Commande

DONE

Implémenter la logique du panier d'achat

DONE

Implémenter la gestion des commandes

DONE

Ajouter la validation des commandes

DONE

Créer les entités Livraison et Service de livraison

DONE

Mettre en place le suivi des livraisons

DONE

Créer l'entité Paiement et configurer le paiement en ligne

DONE

+ Ajouter une carte

Sprint 5 : Interface utilisateur et design

DONE

Implémenter le front-end du site (Bootstrap, responsive design)

DONE

Créer la page d'accueil

DONE

Designer les différentes pages

DONE

Tester la navigation responsive du site

+ Ajouter une carte

Sprint 6 : Tests et déploiement

DONE

Préparer le déploiement (serveur, base de données, SSL, etc.)

DONE

Corriger les bugs et ajustements

DONE

Déployer le projet sur un environnement de production

DONE

Tester l'environnement de production

IN PROGRESS

Trouver les améliorations possible et l'ajout de fonctionnalités





















+ Ajouter une carte



# Github

 <b>RomainRas</b> commit du Mercredi 16 octobre	e4e6f59 · now	 <b>6 Commits</b>
 <b>assets</b>	commit du Mercredi 16 octobre	9 minutes ago
 <b>cahier_des_charges</b>	commit du Mercredi 16 octobre	9 minutes ago
 <b>maquette_et_wireframe/Maquette/Idées Maqu...</b>	commit du Mercredi 16 octobre	9 minutes ago
 <b>mcd_bdd</b>	commit du Mercredi 16 octobre	9 minutes ago
 <b>presentation</b>	commit du Mercredi 16 octobre	now
 <b>README (cahier des charges).md</b>	commit du Mercredi 16 octobre	now

# Github

 <b>RomainRas</b> Update index.html.twig	7e031ca · 2 days ago	 13 Commits
 assets	Initial commit	last week
 bin	Add webapp packages	4 months ago
 config	commit du 22 janvier 2025	last month
 migrations	Initial commit	last week
 public	Initial commit	last week
 src	Initial commit	last week
 templates	Update index.html.twig	2 days ago
 tests	first commit	3 months ago
 translations	Add webapp packages	4 months ago
 .env.test	Initial commit	last week
 .gitignore	Initial commit	last week
 compose.override.yaml	Add webapp packages	4 months ago
 compose.yaml	Add webapp packages	4 months ago
 composer.json	commit du 22 janvier 2025	last month
 composer.lock	commit du 22 janvier 2025	last month
 importmap.php	Add webapp packages	4 months ago
 phpunit.xml.dist	Add webapp packages	4 months ago
 symfony.lock	first commit	3 months ago



# Technologies

# Back-end

- Php ( Framework Symfony )

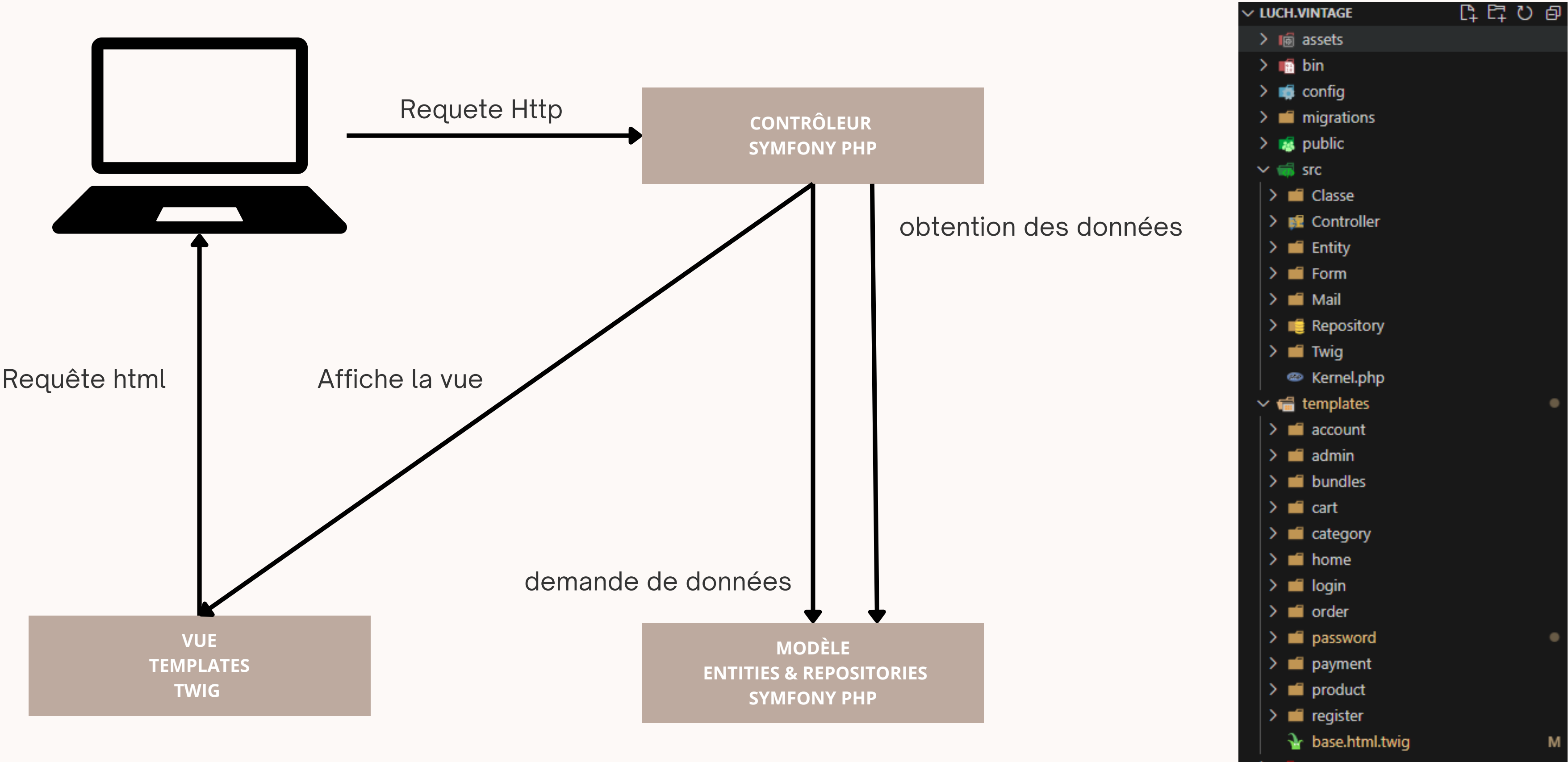
# Front-End

- HTML
- CSS
- JavaScript
- Twig



# Développement

# Architecture ( MVC )





# Exemple de code

Front-End, Back-end, API

## Inscription

- Soumission du formulaire
- Personnalisation du formulaire
- Message Flash et personnalisation des messages flashes
- Mailing

# Soumission du formulaire

RegisterController.php

```
/** -> Route pour l'inscription
#[Route(path: '/inscription', name: 'app_inscription')]

/** ** Méthode pour afficher le formulaire d'inscription et gérer l'enregistrement de l'utilisateur. ** /
20 references | 0 overrides
public function index(Request $request, EntityManagerInterface $entityManager): Response
{
    /** -> Crée une nouvelle instance de l'entité `User`, qui sera remplie avec les données du formulaire.
    $user = new User();

    /** -> Crée un formulaire basé sur `RegisterUserType` et l'associe à l'objet `$user`, pour lier les champs du formulaire aux propriétés de l'entité `User`.
    $form = $this->createForm(type: RegisterUserType::class, data: $user);

    /** -> Verification si le formulaire a bien été soumis
    $form->handleRequest(request: $request);

    /** -> Vérifie que le formulaire a été soumis et que les données sont valides (par exemple, email au bon format, champs requis remplis).
    if ($form->isSubmitted() && $form->isValid()) {

        /** -> Preparation de l'insertion de l'objet `$user` dans la base de données.
        $entityManager->persist(object: $user);

        /** -> Execution de l'insertion, enregistrant ainsi l'utilisateur dans la base de données.
        $entityManager->flush();
```

# Personnalisation du formulaire

RegisterUserType.php

```
/** -> Objet pour ajouter les champs au formulaire
$builder
->add(child: 'email', type: EmailType::class, options: [ // NotBlank, email, UniqueEntity
    'label' => "Votre Adresse Email",
    'attr' => [
        'placeholder' => "Indiquez votre adresse email",
        'class' => 'form-control registration-input'
    ],
    'constraints' => [
        new Assert\NotBlank(options: [
            'message' => "L'adresse email est obligatoire."
        ]),
        new Assert\Email(options: [
            'message' => "L'adresse email '{{ value }}' n'est pas valide."
        ]),
    ],
],
->add(child: 'plainPassword', type: RepeatedType::class, options: [ // NotBlank, min12, regex
    'type' => PasswordType::class,
    'constraints' => [
        new Assert\NotBlank(options: [
            'message' => "Le mot de passe est obligatoire."
        ]),
        new Length(exactly: [
            'min' => 12,
            'max' => 30,
            'minMessage' => "Le mot de passe doit contenir au moins 12 caractères.",
            'maxMessage' => "Le mot de passe ne doit pas dépasser 30 caractères."
        ]),
        new Assert\Regex(pattern: [
            'pattern' => '/^(?=.*[A-Z])(?=.*[\W])(?=.*\d)(?!.*\s){12,30}$/',
            'message' => "Le mot de passe doit contenir au moins une majuscule, un chiffre, un symbole et ne pas contenir d'espace"
        ]),
    ],
    'first_options' => [
        'label' => 'Votre Mot de Passe',
        'attr' => [
            'placeholder' => "Choisissez votre mot de passe",
            'class' => 'form-control registration-input'
        ],
        'hash_property_path' => 'password'
    ],
    'second_options' => [
        'label' => 'Confirmez votre mot de passe',
        'attr' => [
            'placeholder' => "Confirmez votre mot de passe",
            'class' => 'form-control registration-input'
        ],
    ],
    'invalid_message' => "Les mots de passe ne correspondent pas.",
    'mapped' => false,
],
];
```

Register.css

```
/* --- Champs du formulaire --- */
.container-registration form {
    display: flex;
    flex-direction: column;
    gap: 15px;
    text-align: left;
}

.container-registration label {
    font-size: 1rem;
    font-weight: bold;
    color: #6e4d46;
    margin-bottom: 5px;
}

.container-registration input {
    width: 100%;
    padding: 10px;
    border-radius: 8px;
    border: 2px solid #bdaaa5;
    font-size: 1rem;
}

.container-registration input:focus {
    border-color: #6e4d46;
    box-shadow: 0 0 8px rgba(110, 77, 70, 0.3);
}

/* --- Bouton de validation --- */
.container-registration .btn-success {
    background-color: #bdaaa5;
    color: white;
    text-transform: uppercase;
    font-weight: bold;
    border-radius: 50px;
    padding: 12px 24px;
    border: none;
    transition: all 0.3s ease-in-out;
    display: block;
    width: 100%;
    text-align: center;
    margin-top: 15px;
}

.container-registration .btn-success:hover {
    background-color: #a28c86;
    transform: translateY(-2px);
}
```

Inscription

Votre Adresse Email

Indiquez votre adresse email

Votre Mot de Passe

Choisissez votre mot de passe

Confirmez votre mot de passe

Confirmez votre mot de passe

Votre Prénom

Indiquez votre prénom

Votre Nom

Indiquez votre nom

VALIDER

Vous avez déjà un compte ?

Connectez-vous



## RegisterController.php

```

    /** -> Ajout d'un message flash de succès pour informer l'utilisateur que son compte a bien été créé.
    $this->addFlash(
        type: 'success',
        message: "Votre compte à bien été créé. Veuillez vous connecter"
    );

```

## base.html.twig

```

<!-- Conteneur principal qui va regrouper tous les messages flash -->
<div class="flash-container">
    <!-- Première boucle : récupère les différents types de messages flash stockés dans la session -->
    {% for label, messages in app.flashes %}
        <!-- Deuxième boucle : parcourt tous les messages associés au label courant -->
        {% for message in messages %}
            <!-- Affichage du message flash dans une div avec une classe Bootstrap "alert" -->
            <!-- "alert-{{ label }}" permet de définir dynamiquement le style en fonction du type de message -->
            <div class="alert mt-4 alert-{{ label }}">
                <!-- Affichage du contenu du message flash -->
                {{ message }}
            </div>
        {% endfor %}
    {% endfor %}
</div> <!-- Fin du conteneur flash-container -->

```

## base.css

```

/* --- Conteneur des messages flash --- */
.flash-container {
    position: fixed;
    top: 80px;
    left: 50%;
    transform: translateX(-50%);
    z-index: 1050;
    width: 50%;
    max-width: 600px;
}

.alert {
    padding: 15px 20px;
    border-radius: 12px;
    box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.2);
    border: none;
    font-weight: bold;
    text-align: center;
    opacity: 1;
    transition: opacity 0.5s ease-in-out;
    font-family: 'Georgia', serif;
}

.alert-success {
    background-color: #bdaaa5;
    color: white;
}

.alert-danger {
    background-color: #e5745d;
    color: white;
}

.alert-warning {
    background-color: #f1c40f;
    color: black;
}

.alert-info {
    background-color: #7a5e56;
    color: white;
}

.alert .btn-close {
    filter: invert(1);
}

```

Mail.php

```
public function send($to_email, $to_name, $subject, $template, $vars = null): void
{
    /** -> 1. Recuperation du template
    $content = file_get_contents(filename: dirname(path: __DIR__).'./Mail/'.$template);

    /** -> 2. Recuperation des variables facultatives
    if ($vars) {
        foreach($vars as $key=>$var) {
            $content = str_replace(search: '{'.$key.'}', replace: $var, subject: $content);
        }
    }

    /** -> 3. Creation d'un client Mailjet
    $mj = new Client(key: $_ENV['MJ_APIKEY_PUBLIC'], secret: $_ENV['MJ_APIKEY_PRIVATE'], call: true, settings: ['version' => 'v3.1']);

    /** -> 4. Construction du corps de l'email
    $body = [
        'Messages' => [
            [
                'From' => [
                    'Email' => "rascle.romain@outlook.fr",
                    'Name' => "Luch Vintage"
                ],
                'To' => [
                    [
                        'Email' => "$to_email",
                        'Name' => "$to_name"
                    ]
                ],
                'TemplateID' => 6625692,
                'TemplateLanguage' => true,
                'Subject' => "$subject",
                'Variables' => [
                    'content' => $content
                ]
            ]
        ]
    ];

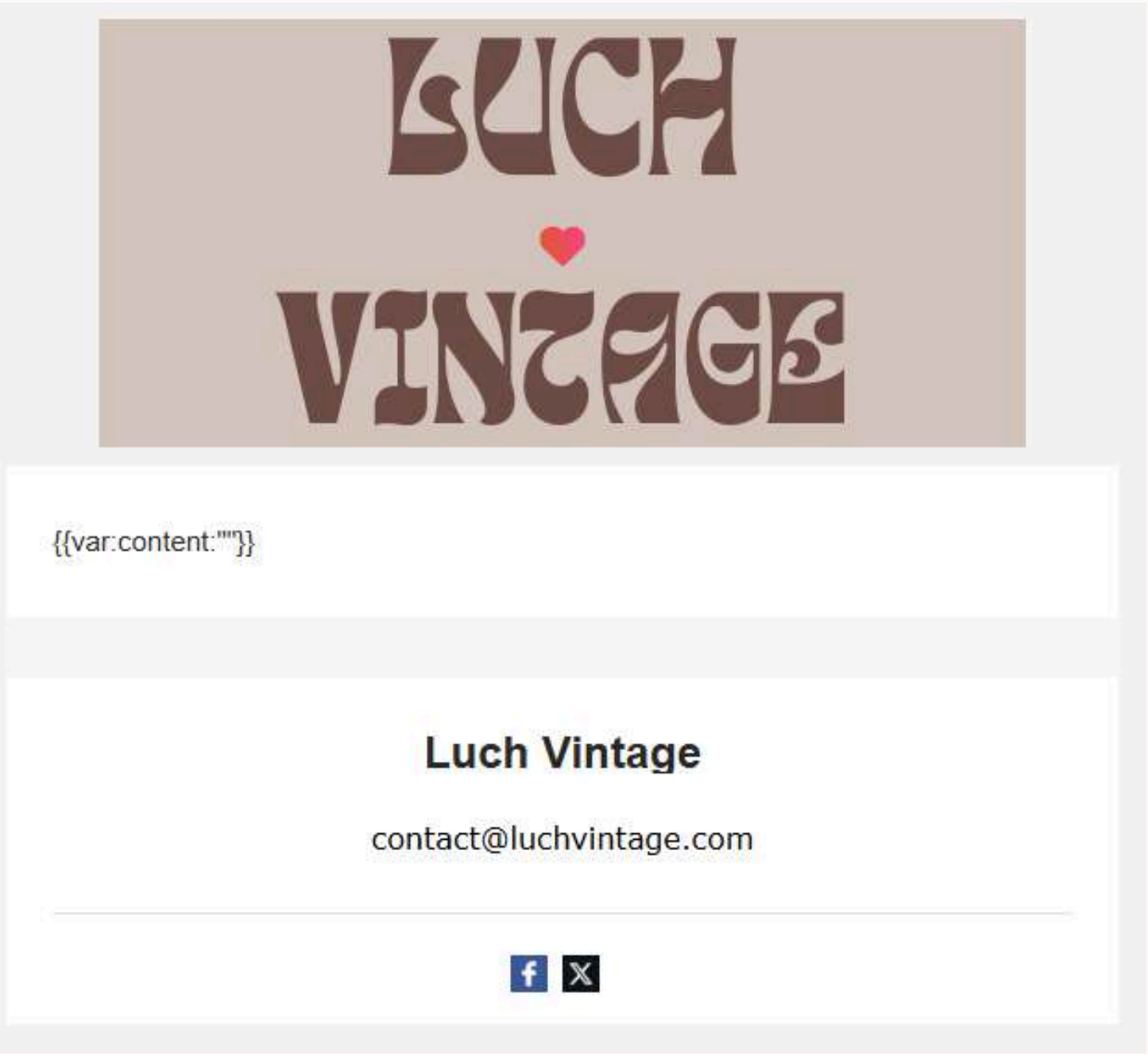
    /** -> 4. Construction du corps de l'email
    $mj->post(resource: Resources::$Email, args: ['body' => $body]);

    /**
     - $mj->post(...) : Envoie une requête POST à l'API Mailjet.
     - Resources::$Email : Ressource de l'API Mailjet utilisée pour l'envoi d'emails.
     - ['body' => $body] : Corps de la requête, qui contient toutes les informations nécessaires pour l'email.
    */
}
```

app.mailjet.com

ID du modèle : 6625692

{{var:content: ""}} : Permet d'ajouter le contenu de welcome.html



# Mailing

## RegisterController.php

```
/* -> Envoie d'un email de confirmation d'inscription
$mail = new Mail();
$vars = [
    'firstname' => $user->getFirstname(),
];
$mail->send(to_email: $user->getEmail(),to_name: $user->getFirstname().' '.$user->getLastname(),subject: 'Bonjour, merci pour votre inscription', template: "welcome.html", vars:...$vars);

/* -> Redirection de l'utilisateur vers la page de connexion après inscription réussie.
return $this->redirectToRoute(route: 'app_login');
```

## welcome.html

```
Bonjour {firstname},<br/><br/>

Merci pour votre inscription sur notre site internet. Bienvenue sur Luch Vintage.<br/>
Vous pouvez dès à présent passer votre commande sur notre plateforme.<br/><br/>
Notre service client reste à votre disposition,<br/>
L'équipe Luch Vintage.
```



# Demonstration

# Parcours utilisateur et Tunnel d'achat

- Inscription
  - Connexion
  - Panier
  - Achat
-

# Fonctionnalités Administrateur

- Création de catégories
  - Ajout d'articles
  - Personnalisation du header
-



# Conclusion

# Amelioration futur

- Wishlist
  - Mailing
  - Article masqué si vendu
  - Optimisation SEO
  - Ajout de plusieurs images par articles
  - Mailing
  - Optimisation SIO ( chargement du site, aria label, balise sémantique )
-



# Conclusion

La création de ce projet m'a permis de :

- Être capable d'apprendre et de travailler en autonomie.
- Résoudre et faire face à chaque problème.
- Avoir la satisfaction de créer quelque chose.

En conclusion, ce projet m'a permis de mettre un premier pied dans le monde du développement et de me donner les armes, la patience et la résilience nécessaires pour continuer à m'épanouir dans ce métier.

**AVEZ-VOUS DES QUESTIONS ?**



# Merci

---