

Dossier-Projet — Refonte du site Decramp

Nom de naissance : Boukhalfa

Prénom : Camil

Adresse : 67 rue Henri Pauquet

Titre professionnel visé :

Développeur web et web mobile Niveau III

Cahier des charges

2. Présentation du projet

Nom du projet : Refonte et modernisation du site web Decramp

Client / Entreprise : Société Decramp (installations et maintenance électriques)

Réalisé par : Boukhalfa Camil

Objectif principal : Actualiser le site web existant pour le rendre conforme aux standards actuels (design, accessibilité, compatibilité mobile, sécurité, interactivité et ergonomie).

3. Contexte du projet

Le site actuel (<https://www.decramp.fr/>) présente l'entreprise et ses services, mais il n'est plus à jour ni conforme aux attentes actuelles du web moderne.

Cette refonte a pour but de :

- Moderniser l'image de la société.
- Rendre la navigation intuitive et agréable sur tous les appareils.
- Ajouter des fonctionnalités interactives (formulaire de contact, dépôt de commentaires, demande de devis, etc.).
- Optimiser les performances et la sécurité.
- Respecter les normes RGPD et d'accessibilité.

- Restreindre certaines interactions : seuls les utilisateurs authentifiés pourront laisser un commentaire ou demander un devis ; les visiteurs non authentifiés pourront toutefois envoyer un message via le formulaire de contact.

4. Objectifs

Objectifs principaux

- Donner une image moderne, claire et professionnelle de la société.
- Faciliter la mise en relation entre clients et entreprise.
- Présenter clairement les services pour **particuliers, professionnels et domotique**.
- Permettre aux visiteurs authentifiés de laisser un avis/commentaire .
- Mettre en valeur les partenaires avec un lien vers leur site.
- Mettre le site aux **normes HTML5 / CSS3 / PHP 8 / RGPD / responsive design**.
- Stocker et classer les images par type (particulier, professionnel, domotique) et permettre l'affichage des 3 dernières images par catégorie.

Objectifs secondaires

- Ajouter des animations discrètes pour renforcer la modernité du site.
- Permettre la création de compte utilisateur.
- Offrir un espace utilisateur avec gestion du profil, historique d'activité et modification d'informations personnelles.
- Intégrer une zone d'administration simple pour consulter et modérer messages, devis et commentaires.
- Implémenter des mesures de sécurité et de performance : HTTPS obligatoire, mots de passe hachés (bcrypt), requêtes préparées (PDO), protection contre XSS/CSRF, sauvegardes régulières et logs.

5. Public cible

_Particuliers : souhaitant des services d'installation, dépannage ou rénovation électrique.

_Professionnels : entreprises, commerces, collectivités cherchant un partenaire fiable pour leurs installations électriques.

_Partenaires techniques : fournisseurs ou sous-traitants potentiels.

_Administrateurs : gestion et modération du contenu (commentaires, demandes de devis, galeries, utilisateurs).

6. Benchmark concurrentiel

Objectifs :

- Analyser les bonnes pratiques actuelles dans le secteur des artisans électriciens.
- Identifier les fonctionnalités attendues par les utilisateurs.
- Identifier les axes d'amélioration et les fonctionnalités différenciantes. .

Éléments analysés :

Critère	LBA Électricien Morangis	Mega Watt – Électricité Alternatif Elec 91	
Design & ergonomie	Basique / ancien	Moderne mais surchargé visuellement	Obsolète / non optimisé
Navigation	Simple mais peu structurée	Moyenne	Simple
Section services	Très limitée	Présente mais peu détaillée	Très limitée
Formulaire de contact	Oui (simple)	Oui	Oui
Avis / témoignages	Non	Oui mais peu mis en avant	Non
Page connexion / compte client	Aucune	Aucune	Aucune
Temps de chargement	Moyen	Rapide	Lent
Qualité du contenu	Minimaliste	Correct	Très pauvre

Responsive / mobile	Partiellement optimisé	Bon	Mauvais
Interactions dynamiques (JS)	Aucune	Très faible	Aucune

KPI	LBA Électricien	Mega Watt	Alternatif Elec 91	Notes générales
Sécurité technique (HTTPS, protections)	Faible	Moyen	Faible	Peu de sites artisans utilisent HTTPS strict + protections anti-injection. Aucun concurrent ne propose de gestion de cookies ou mentions RGPD correctes.
Conformité RGPD	Faible	Faible	Inexistante	Peu de balises meta, contenus courts, mauvaise structure Hn.
SEO global	Faible	Moyen	Faible	
Performance (GTmetrix/Core Web Vitals)	0,6	0,85	0,4	Très variable, Mega Watt étant le plus performant.
Accessibilité (contraste, aria-labels)	Faible	Moyen	Très faible	Aucune optimisation spécifique observée.
Contenu commercial	Minimal	Correct	Insuffisant	Sites pauvres en contenu informatif.
Taux de conversion potentiel	Faible	Moyen	Faible	Peu d'éléments incitatifs (CTA, preuves sociales...).

Résumé du benchmark :

Ce que font les concurrents locaux :

La majorité des sites d'électriciens de Morangis sont :

- Très simples,
- Peu détaillés,
- Souvent anciens ou peu actualisés,
- Sans fonctionnalités avancées,
- Généralement limités à :
 - _Présentation rapide
 - _Contact
 - _Téléphone urgent
 - _Liste des prestations **très courtes**

Aucun ne propose :

- Espace membre,
- Back-office,
- Système de commentaires,
- Protection reCAPTCHA,
- Carrousel moderne,
- Gestion d'images avancée,
- Design professionnel structuré.

7. Analyse SWOT

Analyse stratégique du projet.

Forces (Strengths)	Faiblesses (Weaknesses)
<ul style="list-style-type: none">• Site sécurisé (reCAPTCHA, CSRF, mots de passe hashés).• Architecture structurée, évolutive et sécurisée (PDO, MVC simplifié, validation serveur).• Espace client + espace admin complet.• Commentaires modérés automatiquement.• Identité visuelle moderne, cohérente et adaptée au secteur• SEO optimisé, balises structurées	<ul style="list-style-type: none">• Temps de développement long.• Nécessite une maintenance régulière.• Pas encore de tableau de bord avancé.• Dépendance à une stack PHP traditionnelle (sans framework moderne).• Fonctionnalités nombreuses donc besoin de documentation.• Nécessite un hébergement compatible PHP 8 + extensions PDO

Opportunités (Opportunities)	Menaces (Threats)
<ul style="list-style-type: none">• Ajouter un système de devis complet.• Développer une prise de rendez-vous automatisée.• Développer une app mobile (Android / iOS).• Ajouter un espace client avec suivi de demandes.	<ul style="list-style-type: none">• Concurrence locale forte (prix/délai).• Risques de spam/attaques (atténués par la sécurité).• Changements réglementaires (RGPD, sécurité).• Concurrents qui pourraient moderniser leur site.

Expressions & analyse des besoins

1. Persona (utilisateur type)

Un persona correspond au profil représentatif d'un utilisateur cible de ton site.



Sophie — Client particulier
35 ans — Propriétaire d'une maison

Objectifs

- Trouver un électricien fiable
- Obtenir un devis sans déplacement
- Lire les avis clients

Freins

- Méfiance envers les artisans inconnus
- Peur du coût final

Attentes

- Un site clair, simple, rapide
- Un formulaire de contact rassurant
- Présentation détaillée des services



Michel — Client professionnel
52 ans — Gérant d'entreprise

Objectifs

- Demander un devis pour un local
- Avoir un artisan disponible rapidement

Freins

- Processus lent ou complexe

Attentes

- Formulaire dédié entreprises
- Prise en charge rapide et efficace



Thierry — Administrateur
Gestion du contenu et modération

Objectifs

- Lire les messages reçus
- Modérer les commentaires
- Mettre à jour les services

Besoins

- Interface d'administration claire
- Notifications des nouvelles demandes
- Outils de modération efficaces

Charte graphique & Maquettes

1. Charte graphique (base)

- **Couleurs principales :**
 - Bleu électrique (#004080) → Confiance et professionnalisme
 - Jaune doré (#FFCC00) → Énergie et dynamisme
 - Gris clair (#F5F5F5) → Modernité et lisibilité
- **Typographies :**
 - Titres : Montserrat Bold (weights 700)
 - Textes : Open Sans Regular (weight 400) — variantes 600 pour les boutons/étiquettes
- **Style visuel :**
 - Icônes : SVG monochrome (utiliser currentColor pour permettre recoloration).
 - Formes : bords arrondis (--radius), ombres douces (--shadow).
 - Boutons : CTA principaux bleu électrique avec hover foncé ; accents-hover jaune doré pour variants.
 - Animations : micro-interactions (transform, shadow) au survol — durées via --transition.
- **Accessibilité & contraintes :**
 - Contraste visuel : viser ratio $\geq 4.5:1$ pour textes normaux (WCAG AA).
 - Police lisible, tailles adaptatives, focus visible sur éléments interactifs.
 - Tous les SVG/ICONS doivent posséder aria-hidden ou aria-label selon le contexte.

MOODBOARD



#004080



#FFCC00



#F5F5F5

TITRES

Montserrat
Bold

TEXTES

Open Sans
Regular



STYLE VISUEL



[CONTACT](#)

MODERNES ET
PROFESSIONNELS

SECIC

DECRAMP

Nous vous conseillons et
assurons la sécurité de vos
installations électriques
professionnelles, avec des
solutions adaptées
à votre activité.



Exemple :

The screenshot shows a website with a dark background and a blurred image of a person in a yellow jacket. At the top, there are two yellow buttons with blue text: "En savoir plus". In the center, the word "Accueil" is displayed in large white letters. Below it, the text "Thierry Decramp - SECIC - Artisan électricien" is written in white. Further down, there is a paragraph in white text: "Électricien depuis plus de 15 ans, spécialisé dans les nouvelles technologies et respectueux des normes." At the bottom, there is a blue button with white text that says "Nous contacter".

"qsdqsdqsdq"

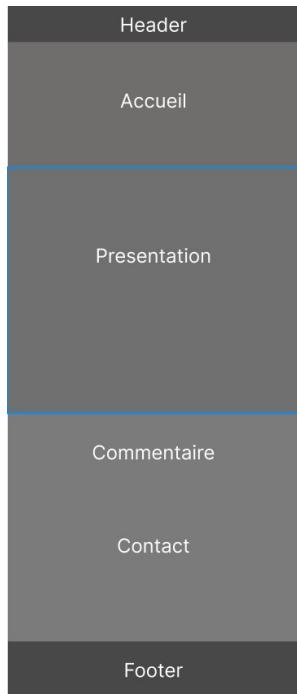
2. Maquette

Pour créer la maquette de mon projet, j'ai opté pour l'utilisation de Figma. J'ai choisi cet outil en raison de sa facilité d'utilisation et de son interface intuitive. Figma offre de nombreuses options qui me permettent de concevoir visuellement l'apparence et la disposition des éléments de mon projet. Je vous présente ici le processus de maquettage du site à partir de l'accueil:

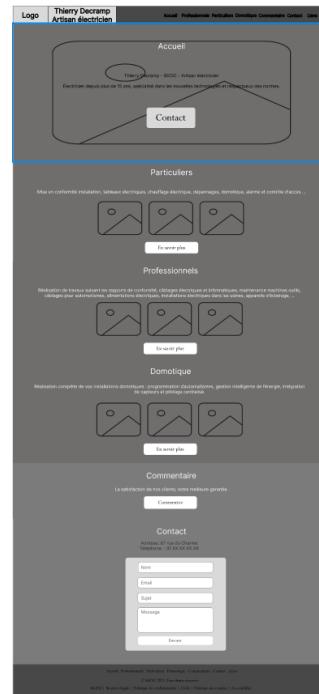
- Zoning : j'ai utilisé l'outil Figma pour reproduire le croquis initial sur un support numérique.
- Wireframe : En utilisant les zones définies dans le zoning, j'ai créé un wireframe basique du de l'accueil. J'ai tracé des rectangles pour représenter chaque zone et j'ai ajouté des étiquettes pour indiquer les différents éléments et quelques images pour indiquer leur emplacement.
- Maquette fonctionnelle : à partir du wireframe, j'ai créé une maquette fonctionnelle de l'accueil. J'ai utilisé des outils de conception tels que des formes, des icônes et des couleurs pour rendre la maquette plus attrayante. J'ai ajouté des éléments réels, tels que le logo du site et les liens de navigation.

Je vous présente ici le résultat du maquettage du site concernant la responsive en utilisant un menu burger (phone et tablette):

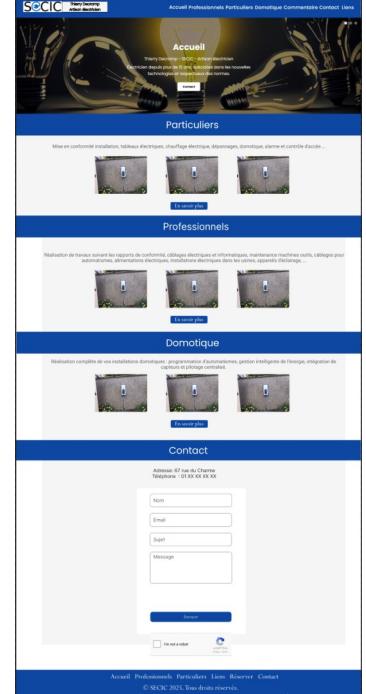
Zoning :



Wireframe :



Maquette:



Responsive : adaptation du menu burger

Lors de la création des maquettes responsive, j'ai conçu une adaptation spécifique du menu de navigation pour les différents formats d'écran (tablette et mobile).

- **Version tablette :**

Le menu burger est positionné **à droite du logo**, dans l'en-tête.

Ce placement permet de conserver une structure proche de la version desktop tout en optimisant l'espace horizontal.

- **Version mobile (téléphone) :**

L'espace disponible étant plus réduit, j'ai déplacé le menu burger **en dessous du logo**.

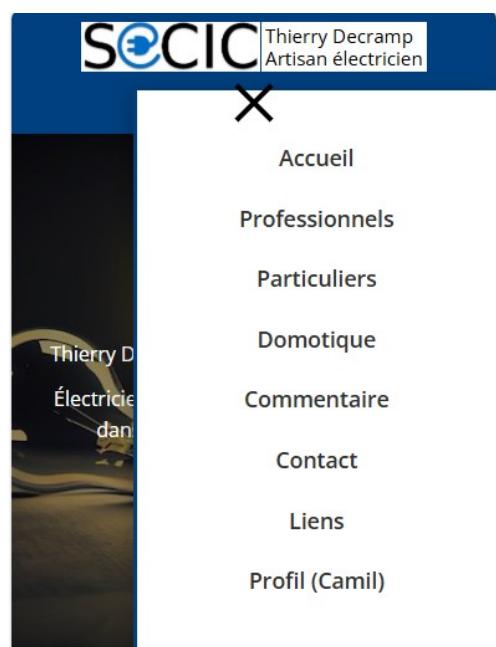
Cette disposition améliore la lisibilité, évite l'encombrement de la barre supérieure et facilite l'accès au menu avec le pouce.

Cette approche garantit une navigation cohérente, intuitive et adaptée à chaque taille d'écran, tout en respectant le principe *mobile-first*.

Tablette :



Phone :



Spécifications

1. Spécifications fonctionnelles

1.1 Fonctionnalités offertes au visiteur

- Consulter la page d'accueil.
- Lire les services et voir leur galeries(particuliers, professionnels, domotique).
- Visualiser les avis.
- Consulter les partenaires et voir leur site.
- Envoyer un message via le formulaire de contact.

1.2 Fonctionnalités offertes à l'utilisateur connecté

- Créer un compte.
- Se connecter.
- Modifier ses informations.
- Déposer un avis .
- Demander un devis.
- Voir l'historique de ses devis et commentaires.

1.3 Fonctionnalités administrateur

- Gestion des utilisateurs.
- Gestion des commentaires (validation, suppression).
- Gestion des devis (lecture, suivi).
- Gestion de la galerie (upload, suppression).
- Gestion des partenaires.
- Consultation des messages reçus.

Parcours utilisateur (User Journey)

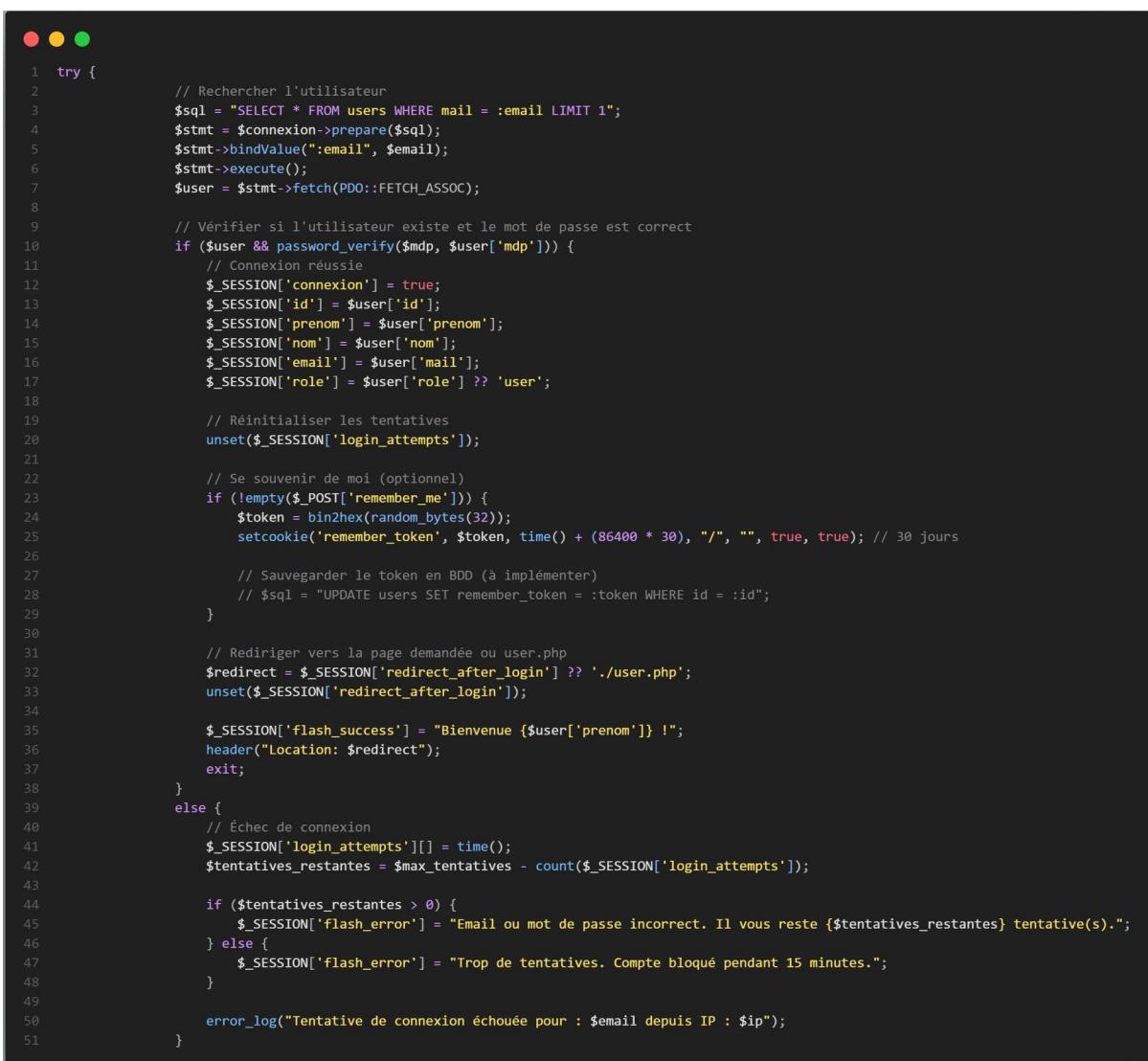
Exemple : visiteur → client

1. Arrive sur la page d'accueil
2. Peut consulter les services : *Particuliers, Professionnels, Domotique.*
3. Vérifie les avis/commentaires
4. Peut consulter les partenaires (avec lien vers leur site).
5. Peut remplir le formulaire de contact
6. L'utilisateur devient client

Utilisateur inscrit

1. Visite la page d'accueil
2. Clique sur Connexion

Explication : Vérification du mot de passe avec `password_verify()`, création de variables de session, comptage des tentatives échouées.



```
try {
    // Rechercher l'utilisateur
    $sql = "SELECT * FROM users WHERE mail = :email LIMIT 1";
    $stmt = $connexion->prepare($sql);
    $stmt->bindValue(":email", $email);
    $stmt->execute();
    $user = $stmt->fetch(PDO::FETCH_ASSOC);

    // Vérifier si l'utilisateur existe et le mot de passe est correct
    if ($user && password_verify($mdp, $user['mdp'])) {
        // Connexion réussie
        $_SESSION['connexion'] = true;
        $_SESSION['id'] = $user['id'];
        $_SESSION['prenom'] = $user['prenom'];
        $_SESSION['nom'] = $user['nom'];
        $_SESSION['email'] = $user['mail'];
        $_SESSION['role'] = $user['role'] ?? 'user';

        // Réinitialiser les tentatives
        unset($_SESSION['login_attempts']);

        // Se souvenir de moi (optionnel)
        if (!empty($_POST['remember_me'])) {
            $token = bin2hex(random_bytes(32));
            setcookie('remember_token', $token, time() + (86400 * 30), "/", "", true, true); // 30 jours
        }

        // Sauvegarder le token en BDD (à implémenter)
        // $sql = "UPDATE users SET remember_token = :token WHERE id = :id";
    }

    // Rediriger vers la page demandée ou user.php
    $redirect = $_SESSION['redirect_after_login'] ?? './user.php';
    unset($_SESSION['redirect_after_login']);

    $_SESSION['flash_success'] = "Bienvenue {$user['prenom']} !";
    header("Location: $redirect");
    exit;
}
else {
    // Échec de connexion
    $_SESSION['login_attempts'][] = time();
    $tentatives_restantes = $max_tentatives - count($_SESSION['login_attempts']);

    if ($tentatives_restantes > 0) {
        $_SESSION['flash_error'] = "Email ou mot de passe incorrect. Il vous reste {$tentatives_restantes} tentative(s).";
    } else {
        $_SESSION['flash_error'] = "Trop de tentatives. Compte bloqué pendant 15 minutes.";
    }

    error_log("Tentative de connexion échouée pour : $email depuis IP : $ip");
}
```

3. Crée un compte via le formulaire d'inscription (sans mail de confirmation)
4. Accède à son espace personnel
5. Modifie ses informations / dépose un commentaire / demande un devis
6. Visualise son activité (messages, commentaires, devis)

Administrateur

1. Se connecte avec rôle admin
2. Accède à une interface privée
3. Gère les devis des utilisateurs.
4. Gère la galerie (ajout/suppression d'images).
5. Gère les partenaires (ajout/modification/suppression).
6. Gère les messages envoyés par les utilisateurs.

Peut modérer / valider / supprimer

1.Arborescence du site

1. Accueil

- Présentation rapide
- Section d'accroche
- Lien vers formulaire de contact
- Boutons d'accès aux sections “Particuliers”, “Professionnels” et “Domotique”
- Animation d'arrière-plan
- Formulaire (nom, prénom, e-mail, sujet, message)
- Envoi via PHP

2. Professionnels

- Description des services proposés
- Galerie d'images
- Lien vers formulaire de contact

3. Particuliers

- Description des services spécialisés

- Galerie d'images
- Lien vers formulaire de contact

4. Domotique

- Description des services spécialisés
- Galerie d'images
- Lien vers formulaire de contact

5. Commentaire

- Système d'avis (Nom prénom de l'utilisateur, note par étoiles, commentaire,date du commentaire)
- Stockage en base de données via PHP/MySQL
- Modération simple (validation de l'admin)
- Lien vers formulaire de contact

6. Contact

- Affichage de disponibilité
- Formulaire (nom, prénom, e-mail, sujet, message)
- Envoi via PHP

7. Liens

- Ressources partenaires et réglementaires (liens vers organismes officiels)

8. Connexion / Inscription

- Formulaire de connexion (email + mot de passe)
- Formulaire d'inscription
- Gestion des erreurs (identifiants invalides, compte inexistant, etc.)
- Redirection vers une autre page pour mot de passe oublié
- Redirection vers l'espace utilisateur après connexion

9. Espace utilisateur (après connexion)

- Tableau de bord
- Modifier les informations personnelles
- Voir les devis envoyés

- Demander un devis
- Laisser un commentaire / voir ses commentaires
- Supprimer son compte (optionnel)

2. Espace administrateur

- Gestion des utilisateurs
- Gestion des commentaires (validation / suppression)
- Gestion des devis (lecture / traitement)
- Gestion de la galerie d'images
- Gestion des partenaires (ajout / modification / suppression)
- Gestion des messages reçus
- Tableau de bord global (statistiques simples)

3. Mentions légales / RGPD

- Informations sur le traitement des données et les droits utilisateurs.

2. Spécifications techniques

1) Architecture du projet

- Organisation PHP classique.
- Séparation front-end / back-end / base de données.

2) Technologies utilisées

- **Front-end :** HTML5, CSS3, JavaScript.
- **Back-end :** PHP 8.
- **Base de données :** MySQL.
- **Outils :** VS Code, XAMPP/WAMP, GitHub, Figma.

3) Sécurité

- Mots de passe hashés (password_hash).
- Requêtes préparées PDO (anti-SQL injection).
- Protection contre XSS.
- Protection CSRF sur les formulaires.
- reCAPTCHA sur formulaire sensible.

- Accès sécurisé (HTTPS).

4) Responsive

- Maquette mobile-first.
- Menu burger :
 - Tablette → à droite du logo
 - Mobile → sous le logo
- Adaptations des sections (images, texte, grille).

5) Gestion des médias

- Upload d'images.

Explication : Vérification du type MIME, taille limitée à 5MB, nom de fichier unique avec `uniqid()`, création du répertoire si inexistant.



```

1 // ===== GESTION DES GALERIES =====
2 if (isset($_POST['upload_image'])) {
3     // retirer récupération service_id
4     $legende = trim($_POST['legende'] ?? '');
5     $image_type = in_array($_POST['image_type'] ?? '', ['particulier','professionnel','domotique'])
6         ? $_POST['image_type'] : 'particulier';
7
8     if (isset($_FILES['image']) && $_FILES['image']['error'] === UPLOAD_ERR_OK) {
9         $allowed_types = ['image/jpeg', 'image/png', 'image/webp', 'image/jpg'];
10        $file_type = $_FILES['image']['type'];
11        $file_size = $_FILES['image']['size'];
12
13        if (!in_array($file_type, $allowed_types)) {
14            $_SESSION['flash_error'] = "Type de fichier non autorisé. Utilisez JPG, PNG ou WebP.";
15        } elseif ($file_size > MAX_FILE_SIZE) {
16            $_SESSION['flash_error'] = "Fichier trop volumineux (max 5 MB).";
17        } else {
18            $extension = pathinfo($_FILES['image']['name'], PATHINFO_EXTENSION);
19            $filename = uniqid('img_', true) . '.' . strtolower($extension);
20
21            if (!is_dir(UPLOAD_DIR)) {
22                mkdir(UPLOAD_DIR, 0755, true);
23            }
24
25            if (move_uploaded_file($_FILES['image']['tmp_name'], UPLOAD_DIR . $filename))
26            {
27                try {
28                    // insertion : filename, legende, image_type, mime_type, file_size
29                    $stmt = $connexion->prepare("INSERT INTO galeries (filename, mime_type, file_size, legende, image_type)
30                                         VALUES (:filename, :mime_type, :file_size, :legende, :image_type)");
31                    $stmt->bindValue(':filename', $filename, PDO::PARAM_STR);
32                    $stmt->bindValue(':mime_type', $file_type, PDO::PARAM_STR);
33                    $stmt->bindValue(':file_size', $file_size, PDO::PARAM_INT);
34                    $stmt->bindValue(':legende', $legende ?: null, $legende === '' ? PDO::PARAM_NULL : PDO::PARAM_STR);
35                    $stmt->bindValue(':image_type', $image_type, PDO::PARAM_STR);
36                    $stmt->execute();
37                    $_SESSION['flash_success'] = "Image uploadée avec succès.";
38                }
39            }
40        }
41    }
42}

```

- Vérification du type MIME.
- Stockage fichier + base de données.

6) Fonctionnalités techniques détaillées

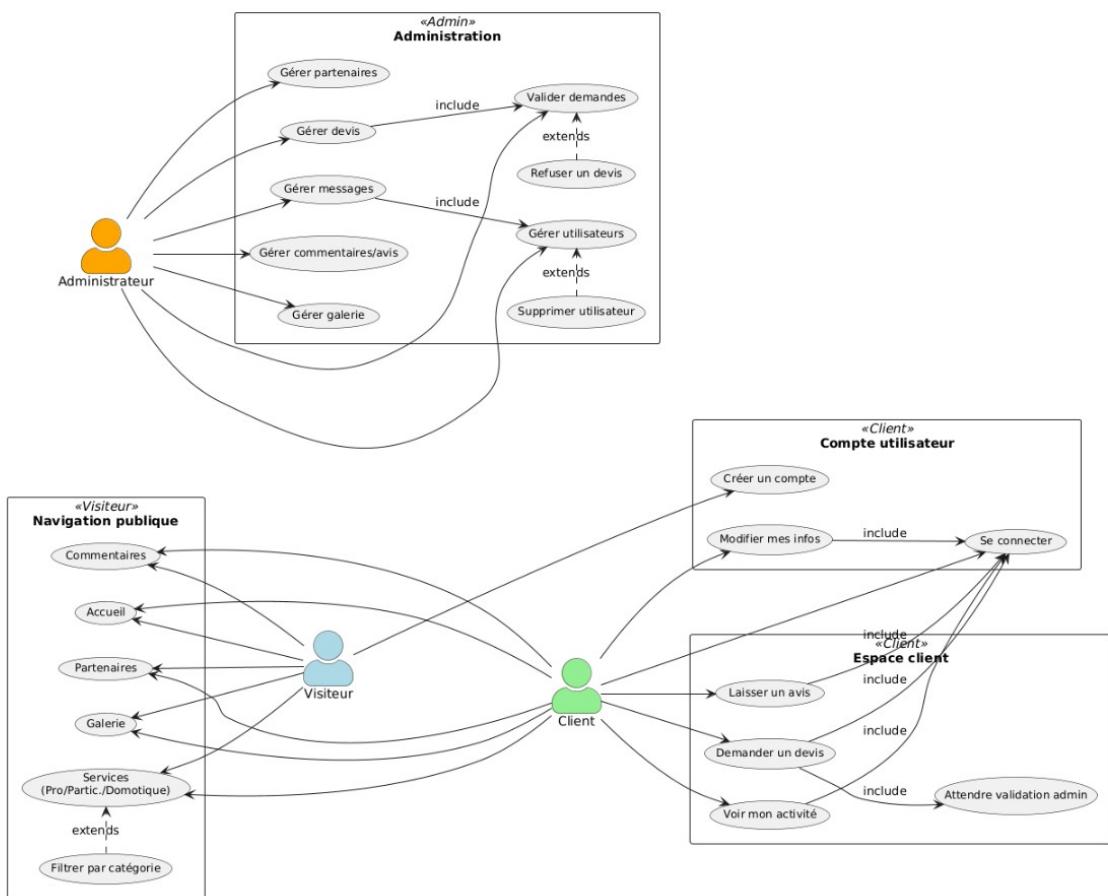
Fonctionnalité	Description
Menu burger responsive	Navigation optimisée pour mobile avec animation “croix”.
Animation Hero	Effet visuel animé en JavaScript pour dynamiser la page d'accueil.
Formulaire de contact	Envoi d'e-mails via PHP (sécurisé, avec validation et messages d'erreur).
Section Commentaires	Enregistrement et affichage des avis utilisateurs avec notation en étoiles.
Galeries d'images	Uniformisation des images avec taille et effet d'ombre.
Responsive Design	Adaptation à tous les écrans (mobile, tablette, ordinateur).
Sécurité PHP	Validation et nettoyage des entrées (prévention XSS et injection SQL).
Accessibilité	Contraste suffisant, balises alt, structure HTML sémantique.
Espace utilisateur	Gestion du compte, historique, devis, commentaires.
Espace administrateur	Interface de gestion complète en PHP/PDO.
Système de devis	Formulaire sécurisé, enregistrement en base, suivi du côté admin et utilisateur
Partenaires	Ajout dynamique depuis l'admin avec un lien cliquable vers leur site
Système de connexion / inscription	Hashage (password_hash) et vérification PDO sécurisée
Modération des avis	Validation par l'admin.
Upload d'images	Stockage en base (nom du fichier, type, légende, service) et vérification.

7) Normes et contraintes

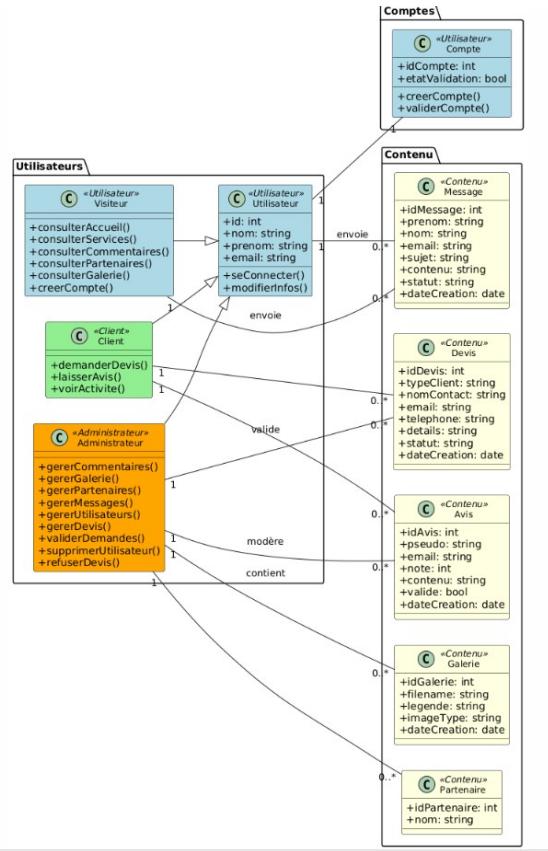
- Respect du RGPD.
 - Normes HTML5 / CSS3.
 - Accessibilité : alt, aria-label, contraste.
 - Compatibilité navigateurs.
 - SEO : meta, structure Hn, sitemap éventuel.
-

Conception UML

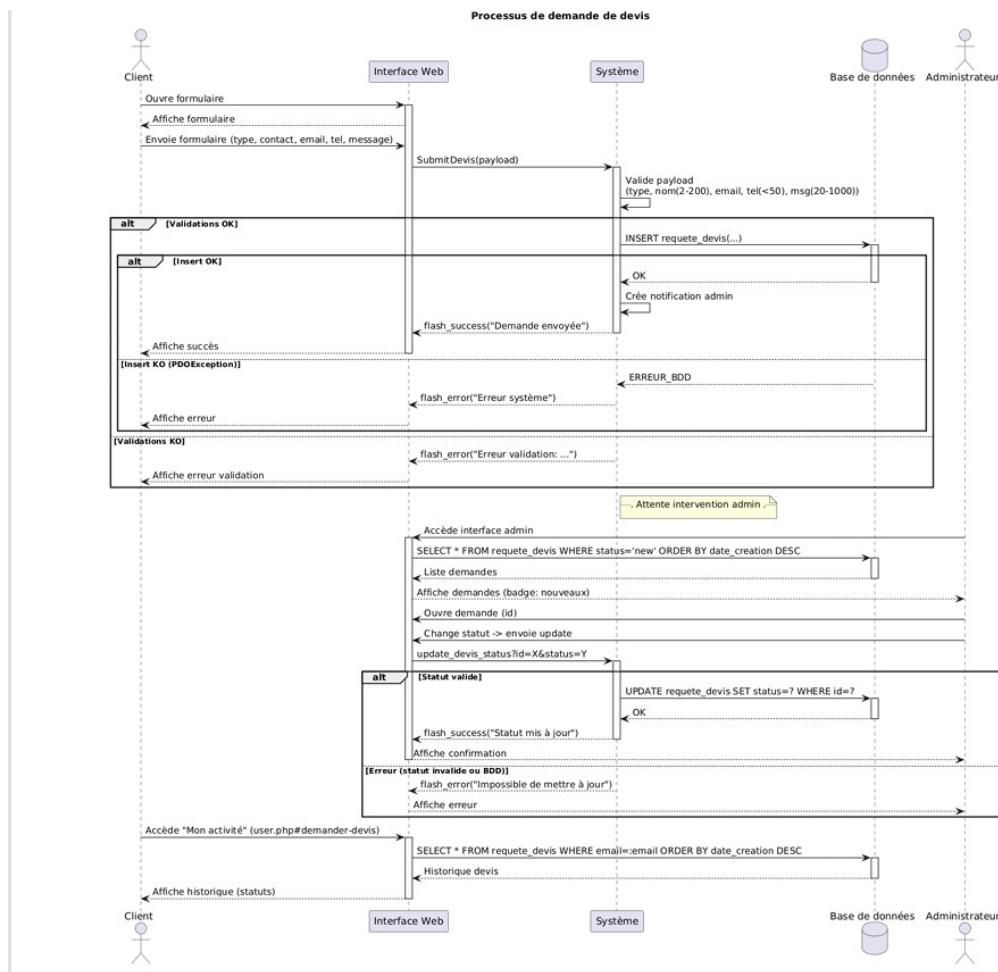
UML-Use Case

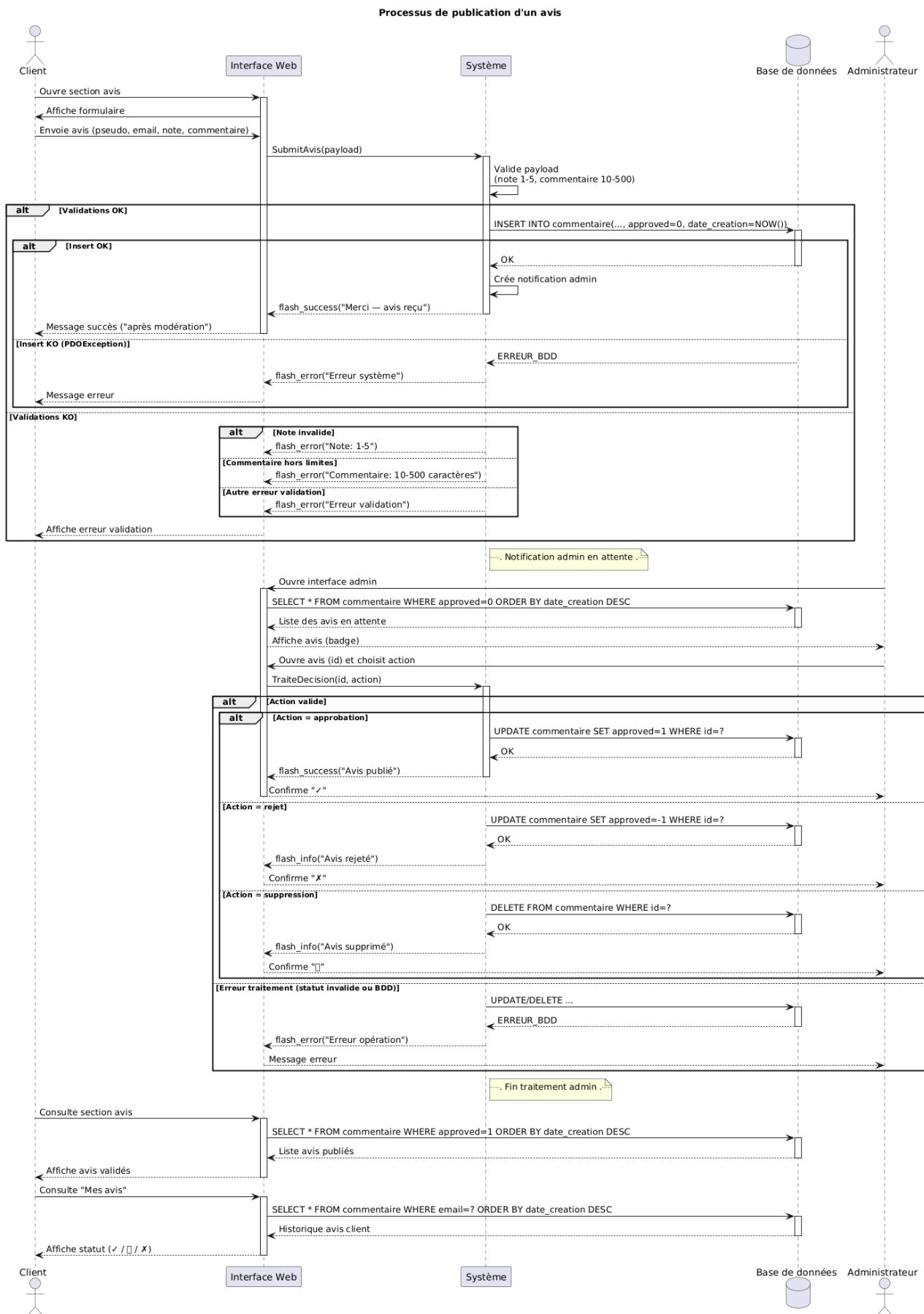


UML — Diagramme de classes



UML — Diagrammes de séquence





MÉTHODE MERISE - BASE DE DONNÉES

1. DICTIONNAIRE DE DONNÉES (DD)

Table: USERS

Description : Comptes utilisateurs inscrits (clients et administrateurs) Les visiteurs non inscrits n'apparaissent PAS dans cette table.

Attribut	Type	Contraintes / Description
id	INT AUTO_INCREMENT	PK
civilite	VARCHAR(3) NULL	M., Mme, Mx
prenom	VARCHAR(50) NOT NULL	
nom	VARCHAR(50) NOT NULL	
mail	VARCHAR(100) NOT NULL UNIQUE	CHECK mail
mdp	VARCHAR(255) NOT NULL	Hash password_hash()
role	VARCHAR(20) DEFAULT 'user'	CHECK role IN ('user','admin')
reset_token	VARCHAR(64) NULL	
reset_token_expires	DATETIME NULL	
date_creation	DATETIME DEFAULT CURRENT_TIMESTAMP	

Table: GALERIES

Description : Images uploadées pour la galerie du site. Peut être liée à une demande de devis (optionnel).

Attribut	Type	Contraintes / Description
id	INT AUTO_INCREMENT	PK
file_name	VARCHAR(255) NOT NULL	Nom du fichier
mime_type	VARCHAR(100) NULL	
file_size	INT UNSIGNED NULL	Taille en octets
image_data	LONGBLOB NULL	
legende	VARCHAR(255) NULL	
image_type	ENUM('particulier','professionnel','domotique') DEFAULT 'particulier'	
date_creation	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	

Table: CONTACT

Description : Messages envoyés via le formulaire de contact. Peut provenir de visiteurs (non inscrits) ou utilisateurs inscrits. Lien implicite avec users via email.

Attribut	Type	Contraintes / Description
id	INT AUTO_INCREMENT	PK
prenom	VARCHAR(100) NULL	
nom	VARCHAR(100) NULL	
email	VARCHAR(255) NOT NULL	CHECK email
sujet	VARCHAR(255) NULL	
message	TEXT NOT NULL	CHECK LENGTH(message) ≥ 10
status	ENUM('new','read','closed') DEFAULT 'new'	
date_creation	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	

Table: COMMENTAIRE

Description : Avis laissés par les utilisateurs inscrits. Modération par admin obligatoire avant publication.

Attribut	Type	Contraintes / Description
id	INT AUTO_INCREMENT	PK
pseudo	VARCHAR(100) NOT NULL DEFAULT	Affiché publiquement
email	VARCHAR(255) NULL	Pas de FK
note	TINYINT UNSIGNED NOT NULL DEFAULT 5	CHECK note BETWEEN 1 AND 5
commentaire	TEXT NULL	CHECK LENGTH(commentaire) 10–500
approved	TINYINT(1) DEFAULT 0	CHECK approved IN (-1, 0, 1)
date_creation	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	

Table: PARTENAIRE

Description : Liste des partenaires affichés sur le site (avec logo/lien).

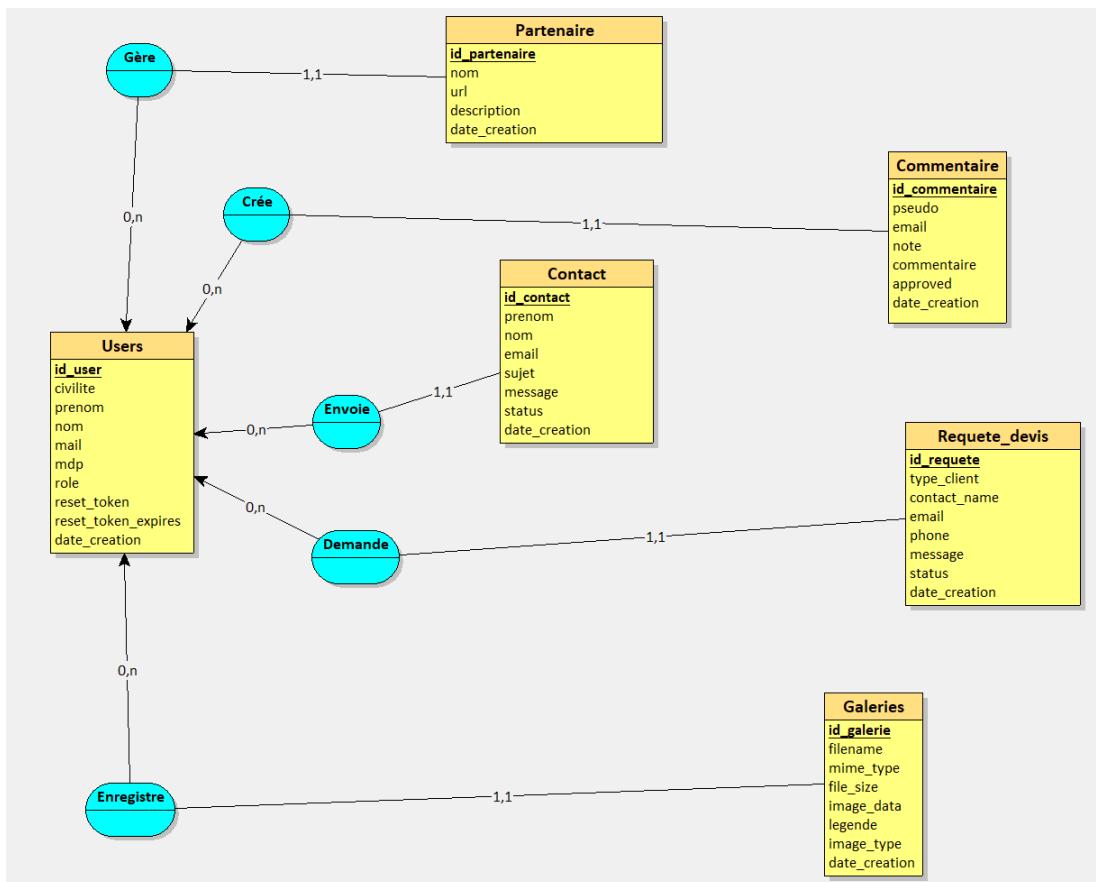
Attribut	Type	Contraintes / Description
id	INT AUTO_INCREMENT	PK
nom	VARCHAR(150) NOT NULL	
url	VARCHAR(255) NOT NULL	CHECK url
description	VARCHAR(255) NULL	
date_creation	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	

Table: REQUETE_DEVIS

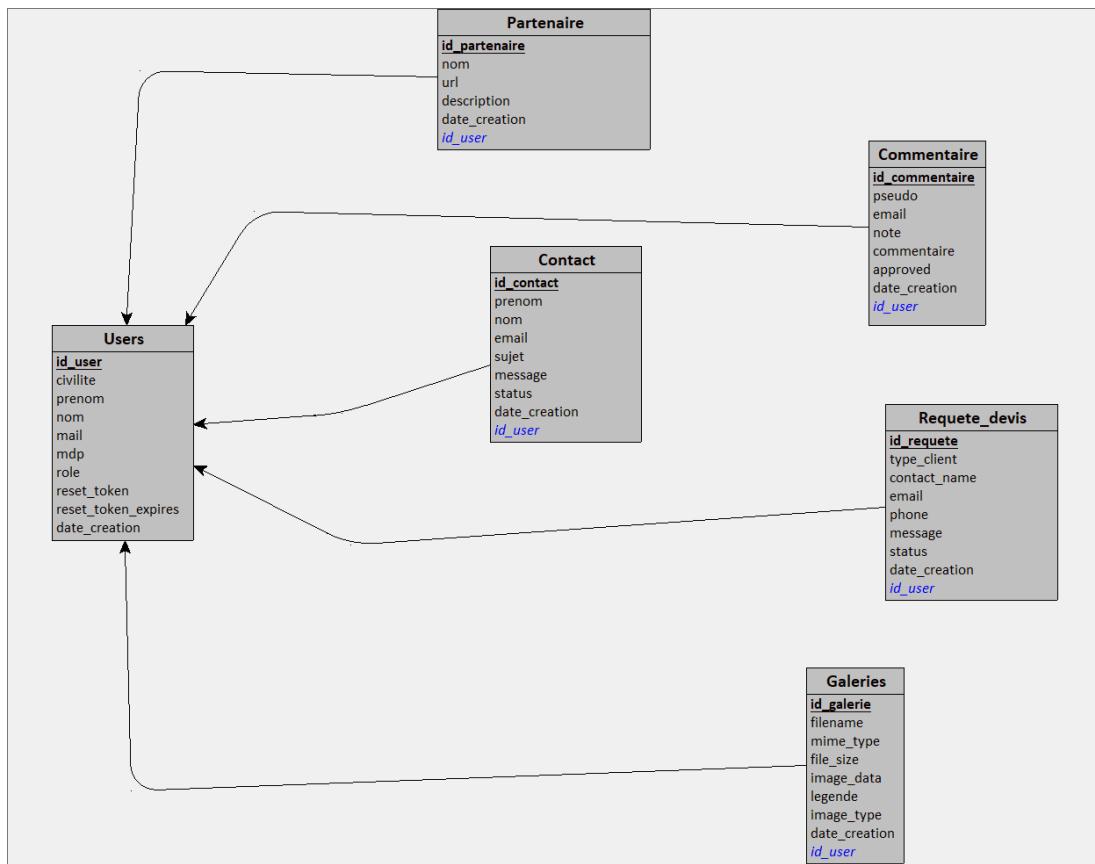
Description : Demandes de devis (clients inscrits). Peut être liée à des images via galeries.requeste_id.

Attribut	Type	Contraintes / Description
id	INT AUTO_INCREMENT	PK
type_client	VARCHAR(200) NULL	CHECK type_client IN ('Particulier','Professionnel','Domotique')
contact_name	VARCHAR(200) NULL	CHECK LENGTH(contact_name) 2–200
email	VARCHAR(255) NOT NULL	
phone	VARCHAR(50) NULL	
message	TEXT NULL	CHECK LENGTH(message) 20–1000
status	ENUM('new','in_progress','closed') DEFAULT 'new'	
date_creation	TIMESTAMP DEFAULT CURRENT_TIMESTAMP	

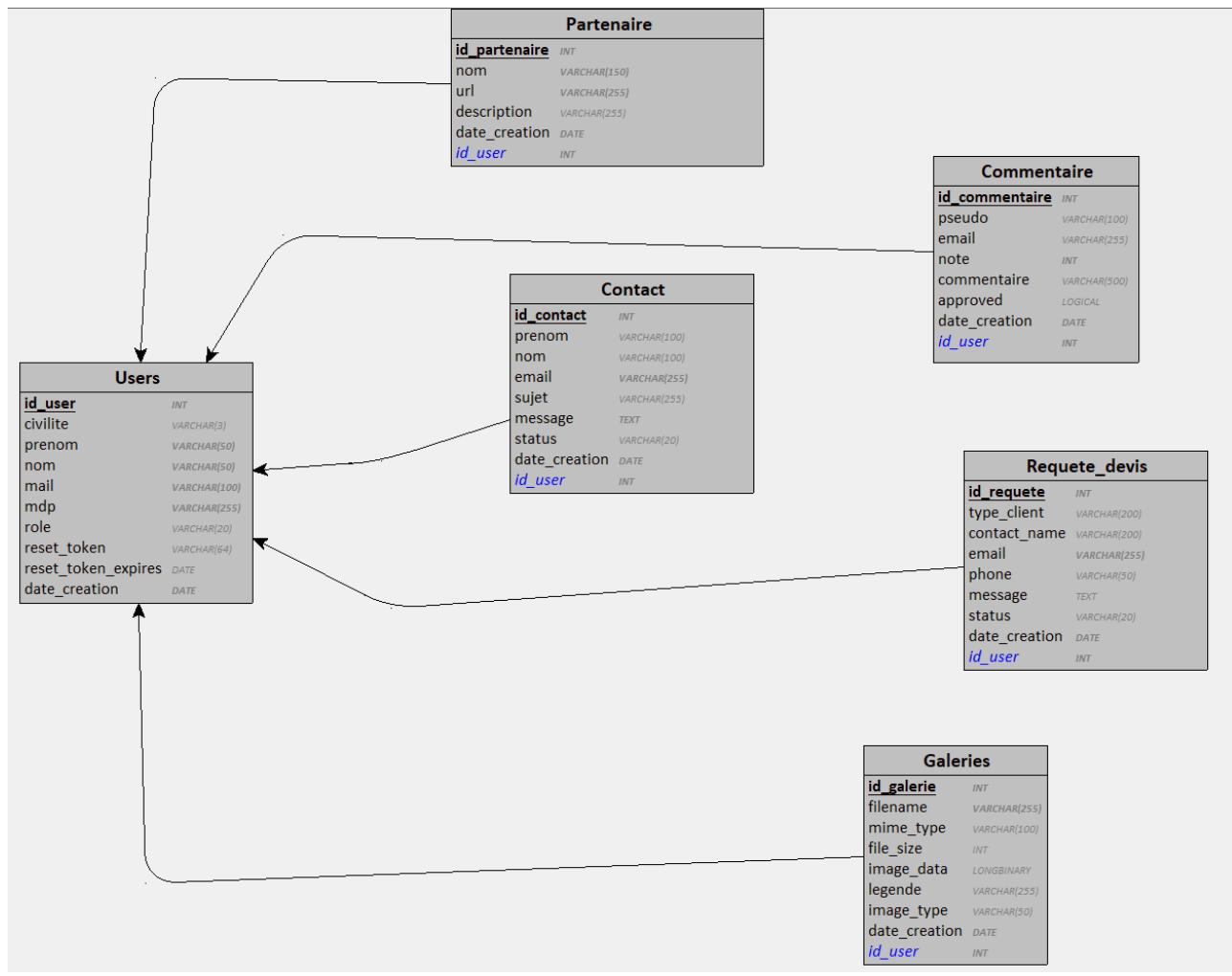
MODÈLE CONCEPTUEL DE DONNÉES (MCD)



MODÈLE LOGIQUE DE DONNÉES (MLD)



MODÈLE PHYSIQUE DE DONNÉES (MPD)



Gestion de projet

Méthode choisie : Agile (Scrum léger)

- Sprints de 2 semaines.
- Réunions courtes , revue de sprint et rétrospective à la fin de chaque sprint.
- Product Owner : Boukhalfa Camil.
- Scrum Master léger : auto-organisation

Cahier des user stories

1. En tant que visiteur, je veux consulter la page d'accueil pour connaître les services.
2. En tant que visiteur, je veux envoyer un message via le formulaire de contact afin de contacter l'entreprise.

3. En tant qu'utilisateur inscrit, je veux créer un compte pour accéder à mon espace personnel.
4. En tant qu'utilisateur authentifié, je veux demander un devis pour lancer une demande.
5. En tant qu'utilisateur authentifié, je veux laisser un commentaire pour partager mon avis.
6. En tant qu'admin, je veux modérer (valider/rejeter) les commentaires.
7. En tant qu'admin, je veux gérer les demandes de devis.
8. En tant que visiteur, je veux voir la galerie d'images par catégorie.
9. En tant qu'utilisateur, je veux modifier mon profil pour corriger mes informations.
10. En tant qu'utilisateur, je veux récupérer mon mot de passe via un email sécurisé.

Backlog & priorisation

Priorité haute (P0) : Auth (inscription/connexion), Contact form, Demande devis, Modération commentaires, Sécurité (HTTPS, hachage mdp, PDO).

Priorité moyenne (P1) : Galerie images, Profil utilisateur, Page partenaires.

Priorité basse (P2) : Animations, micro-interactions, tableaux de bord avancés.

Planning des sprints (exemple 3 sprints)

- Sprint 1 (2 sem) : Auth, DB schema, Contact form, déploiement local.
- Sprint 2 (2 sem) : Demande devis (form + storage + admin), commentaires (soumission + modération).
- Sprint 3 (2 sem) : Galerie, partenaires, responsive polishing, tests et corrections.

Outils & collaboration

- Notion / Trello : backlog + tickets (cards), définir labels (ToDo/InProgress/Done).
 - Discord : communication / partage d'écrans / logs.
 - Git & GitHub : versionning, branches feature/bugfix, PR pour revue.
 - Documentation : dossier pro (livrables).
-

Développement

Environnement de travail

- Éditeur : Visual Studio Code (+ extensions PHP Intelephense, PHP Debug, Prettier).
- Serveur local : Wampserver / XAMPP (PHP 8+, MySQL).
- Déploiement : hébergement LAMP ou VPS.

Mise en place BDD (extrait SQL de création)

```
1 //users
2
3 $sql = "CREATE TABLE IF NOT EXISTS users(
4     id INT AUTO_INCREMENT PRIMARY KEY,
5     civilite VARCHAR(3),
6     prenom VARCHAR(50) NOT NULL,
7     nom VARCHAR(50) NOT NULL,
8     mail VARCHAR(100) UNIQUE NOT NULL,
9     mdp VARCHAR(255) NOT NULL,
10    `role` VARCHAR(20),
11    reset_token VARCHAR(64) DEFAULT NULL,
12    reset_token_expires DATETIME DEFAULT NULL,
13    date_creation DATETIME DEFAULT CURRENT_TIMESTAMP
14 ) CHARACTER SET utf8 COLLATE utf8_bin";
15
16 $connexion->exec($sql);
```

Bonnes pratiques de développement

- Utiliser PDO avec requêtes préparées (prévention SQL injection).
- Hacher les mots de passe avec `password_hash($pwd, PASSWORD_DEFAULT)` + `password_verify`.
- Escaper toute sortie HTML (escape via `htmlspecialchars`).
- Organiser le code en fichiers: /public (front), /src (logique), /templates, /config, /tests.

Exemple d'utilisation PDO + password_hash

Explication : Le mot de passe est haché avec `PASSWORD_DEFAULT` avant stockage. La requête préparée avec `bindValue` protège contre les injections SQL.

```
1 // Créer le compte
2         $mdp_hash = password_hash($mdp, PASSWORD_DEFAULT);
3
4         $sql = "INSERT INTO users (civilite, prenom, nom, mail, mdp, role, date_creation)
5                 VALUES (:civilite, :prenom, :nom, :email, :mdp, 'user', NOW())";
6         $stmt = $connexion->prepare($sql);
7         $stmt->bindValue(":civilite", $civilite);
8         $stmt->bindValue(":prenom", $prenom);
9         $stmt->bindValue(":nom", $nom);
10        $stmt->bindValue(":email", $email);
11        $stmt->bindValue(":mdp", $mdp_hash);
12        $stmt->execute();
```

Explication : Utilisation de `prepare()` et `bindValue()` pour séparer la requête SQL des données. Le `try-catch` capture les erreurs de BDD.

```
1 try {
2     $stmt = $connexion->prepare("INSERT INTO partenaire (nom, `url`, `description`) VALUES (:nom, :url, :description)");
3     $stmt->bindValue(':nom', $nom, PDO::PARAM_STR);
4     $stmt->bindValue(':url', $url, PDO::PARAM_STR);
5     $stmt->bindValue(':description', $description ?: null, $description === '' ? PDO::PARAM_NULL : PDO::PARAM_STR);
6     $stmt->execute();
7     $_SESSION['flash_success'] = "Partenaire ajouté avec succès.";
8 } catch (PDOException $e) {
9     $_SESSION['flash_error'] = "Erreur lors de l'ajout du partenaire.";
10    error_log("Erreur ajout partenaire : " . $e->getMessage());
11 }
```

Template bug

- ID : BUG-001
- Titre : Formulaire contact n'envoie pas d'email
- Environnement : Wampserver, PHP 8.1
- Reproduction : étapes ...
- Correction : utiliser PHPMailer + config SMTP, solution appliquée ...
- Statut : Résolu



SEO (optimisation pour les moteurs)

Exemple de <title> et <meta>

```
<meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="description" content="Thierry Decramp - Artisan électricien à
L'Isle-Adam. Plus de 15 ans d'expérience en électricité, domotique et installations
professionnelles.">
  <meta name="keywords" content="électricien, domotique, installation électrique,
L'Isle-Adam, Val d'Oise">
  <meta name="author" content="SECIC - Thierry Decramp">
  <title>Accueil - <?php echo SITE_TITLE; ?></title>
```

Checklist SEO on-page (à inclure dans ton dossier)

- **Pages principales** : <title> unique, <meta description> optimisée (120–160 caractères).
- **Hn structure logique** : H1 unique par page, H2/H3 pour sous-sections.
- **URLs propres et lisibles** : ex : /services/particuliers, /domotique.
- **Balises alt descriptives sur toutes les images** pour accessibilité et SEO.
- **Sitemap XML généré et soumis** à Google Search Console.
- **robots.txt configuré** pour autoriser indexation pages publiques et bloquer /admin.
- **Structured Data JSON-LD** pour LocalBusiness et avis si pertinent.

RGPD & Sécurité

Actions obligatoires / checklist RGPD

- **Politique de confidentialité** claire (qui, quoi, pourquoi, durée).
- **Consentement cookies** (bannière + gestion granulaire).
- **Formulaire** : case à cocher pour accepter la politique de confidentialité si données personnelles collectées.

Explication : L'inscription est bloquée si l'utilisateur n'accepte pas les CGU.
Conforme au RGPD (consentement explicite).

```
● ● ●
1 elseif (!$accepter_cgus) {
2     $_SESSION['flash_error'] = "Vous devez accepter les conditions générales d'utilisation.";
3 }
```

```
● ● ●
1 <div class="checkbox-group">
2     <input type="checkbox" id="accepter-cgu" name="accepter_cgus"
3         value="1" required>
4     <label for="accepter-cgu">
5         J'accepte les <a href="./cgus.php" target="_blank">conditions générales d'utilisation</a> *
6     </label>
7 </div>
```

- **Droit d'accès / suppression** : procédure pour recevoir et traiter les demandes (email + délai).

Explication : Suppression complète des données personnelles (utilisateur + commentaires + messages). Implémente le droit à l'effacement du RGPD.

```
● ● ●
1 if (!empty($_POST["supprimer_compte"]))
2 {
3     $confirmation = $_POST["confirmation"] ?? '';
4
5     if ($confirmation === 'SUPPRIMER')
6     {
7         try
8         {
9             // Supprimer les données associées (optionnel)
10            $connexion->prepare("DELETE FROM commentaire WHERE email = ?")->execute([$user['mail']]);
11            $connexion->prepare("DELETE FROM contact WHERE email = ?")->execute([$user['mail']]);
12
13            // Supprimer le compte
14            $stmt = $connexion->prepare("DELETE FROM users WHERE id = ?");
15            $stmt->execute([$user_id]);
16
17            // Détruire la session
18            session_destroy();
19
20            header("Location: ./index.php");
21            exit;
22        }
23        catch(PDOException $e)
24        {
25            error_log("Erreur suppression compte : " . $e->getMessage());
26            $_SESSION['flash_error'] = "Erreur lors de la suppression du compte.";
27        }
28    }
29    else
30        $_SESSION['flash_error'] = "Confirmation incorrecte. Tapez 'SUPPRIMER' pour confirmer.";
31    header("Location: ./user.php#profil");
32    exit;
33 }
```

- **Durée de conservation** : définir périodes (ex : logs 90j, devis 2 ans, utilisateurs actifs durée nécessaire).
- **Sécurisation** : HTTPS, hachage mdp, protection CSRF (token), validation côté serveur, limitation taille fichiers uploadés, vérification MIME.

- **Registre des activités** (documenter traitement des données).

Template clause

Les données collectées via les formulaires (nom, email, message) sont utilisées uniquement pour traiter votre demande. Elles sont conservées pendant [durée] et ne sont pas partagées à des tiers. Vous pouvez demander l'accès, la modification ou la suppression de vos données à l'adresse : contact@decramp.fr.

Sécurité

- Forcer HTTPS.

Explication : Vérification côté serveur du reCAPTCHA v2 de Google.
Protection contre les bots et le spam sur le formulaire de contact.

```

1 // ===== VÉRIFICATION reCAPTCHA =====
2 $recaptcha_response = $_POST['g-recaptcha-response'] ?? '';
3
4 if (empty($recaptcha_response)) {
5     $_SESSION['flash_error'] = "Veuillez cocher la case reCAPTCHA.";
6     header("Location: ./contact.php");
7     exit;
8 }
9
10 // Vérifier le reCAPTCHA côté serveur
11 $recaptcha_url = 'https://www.google.com/recaptcha/api/siteverify';
12 $recaptcha_data = [
13     'secret' => RECAPTCHA_SECRET_KEY,
14     'response' => $recaptcha_response,
15     'remoteip' => $_SERVER['REMOTE_ADDR']
16 ];
17
18 $recaptcha_options = [
19     'http' => [
20         'method' => 'POST',
21         'header' => 'Content-Type: application/x-www-form-urlencoded',
22         'content' => http_build_query($recaptcha_data),
23         'timeout' => 10
24     ]
25 ];
26
27 $recaptcha_context = stream_context_create($recaptcha_options);
28 $recaptcha_result = @file_get_contents($recaptcha_url, false, $recaptcha_context);
29
30 if ($recaptcha_result === false) {
31     error_log("Erreur connexion reCAPTCHA");
32     $_SESSION['flash_error'] = "Erreur de validation reCAPTCHA. Veuillez réessayer.";
33     header("Location: ./contact.php");
34     exit;
35 }
36
37 $recaptcha_json = json_decode($recaptcha_result);
38
39 if (!$recaptcha_json || !isset($recaptcha_json->success) || !$recaptcha_json->success) {
40     $error_codes = isset($recaptcha_json->{'error-codes'}) ? implode(', ', $recaptcha_json->{'error-codes'}) : 'inconnu';
41     error_log("reCAPTCHA failed - Error codes: " . $error_codes);
42     $_SESSION['flash_error'] = "Validation reCAPTCHA échouée. Veuillez réessayer.";
43     header("Location: ./contact.php");
44     exit;
45 }

```

- Headers de sécurité : Content-Security-Policy, X-Frame-Options: DENY, X-Content-Type-Options: nosniff, Referrer-Policy.
- Limiter tentatives de connexion (rate limiting) et verrouillage temporaire.
Explication : Bloque l'IP après 5 tentatives échouées pendant 15 minutes.
Stockage en session pour éviter les attaques par dictionnaire.



```

1 // Protection contre le bruteforce : limiter les tentatives
2     $ip = $_SERVER['REMOTE_ADDR'];
3     $max_tentatives = 5;
4     $delai_blocage = 900; // 15 minutes en secondes
5
6     // Vérifier les tentatives de connexion
7     if (!isset($_SESSION['login_attempts'])) {
8         $_SESSION['login_attempts'] = [];
9     }
10
11    // Nettoyer les anciennes tentatives
12    $_SESSION['login_attempts'] = array_filter($_SESSION['login_attempts'], function($time) use ($delai_blocage) {
13        return (time() - $time) < $delai_blocage;
14    });
15
16    // Vérifier si trop de tentatives
17    if (count($_SESSION['login_attempts']) >= $max_tentatives) {
18        $temps_restant = ceil(($delai_blocage - (time() - min($_SESSION['login_attempts']))) / 60);
19        $_SESSION['flash_error'] = "Trop de tentatives de connexion. Veuillez réessayer dans {$temps_restant} minutes.";
20    }

```

- Sauvegardes régulières de la BDD (planifier et décrire le process).

Tests & Déploiement

Plan de tests (types)

- Tests fonctionnels (toutes les user stories).

Explication : *Chaque champ est validé côté serveur avec des règles métier. Les données erronées sont renvoyées au formulaire pour correction.



```
1 // Validation des données
2 if (strlen($nom) < 2 || strlen($nom) > 100) {
3     $_SESSION['flash_error'] = "Le nom doit contenir entre 2 et 100 caractères.";
4     $_SESSION['form_data'] = $_POST;
5     header("Location: ./contact.php");
6     exit;
7 }
8
9 if (!filter_var($mail, FILTER_VALIDATE_EMAIL)) {
10     $_SESSION['flash_error'] = "L'adresse email n'est pas valide.";
11     $_SESSION['form_data'] = $_POST;
12     header("Location: ./contact.php");
13     exit;
14 }
15
16 if (strlen($sujet) < 3 || strlen($sujet) > 200) {
17     $_SESSION['flash_error'] = "Le sujet doit contenir entre 3 et 200 caractères.";
18     $_SESSION['form_data'] = $_POST;
19     header("Location: ./contact.php");
20     exit;
21 }
22
23 if (strlen($message) < 10 || strlen($message) > 1000) {
24     $_SESSION['flash_error'] = "Le message doit contenir entre 10 et 1000 caractères.";
25     $_SESSION['form_data'] = $_POST;
26     header("Location: ./contact.php");
27     exit;
28 }
```

- Tests unitaires (PHP : PHPUnit si utilisé).
- Tests d'intégration (login -> actions utilisateur).
- Tests E2E (cypress / puppeteer si possible).
- Tests sécurité (scan vulnérabilités, check injection, XSS).
- Tests de charge basiques (abrégé) si possible.

Cas de test exemple (format)

- ID : TC-01
- Titre : Inscription utilisateur valide

- Préconditions : page signup accessible
- Étapes : remplir champs valides -> soumettre
- Résultat attendu : redirection dashboard, entrée en BDD, mail (si implémenté)

Déploiement

1. Préparer branche `main` stable.
2. Build / minify assets (CSS/JS).
3. Sauvegarde BDD actuelle.
4. Transfert fichiers via Git/FTP/SSH.
5. Exécuter migrations SQL si nécessaire.
6. Vérifier permissions dossiers (uploads).
7. Tester pages critiques (login, contact, formulaire devis, galerie).
8. Mettre à jour sitemap et vérifier absence d'erreurs 500.

Jeu d'essai

- Users : user1@test.com / password123 (haché), admin@test.com / adminPass
 - Requête devis : 3 demandes (particulier/professionnel/domotique)
 - Commentaires : 5 avis (mix approuvé/en attente/rejeté)
 - Galeries : 6 images (2 par catégorie)
- Documentez ces jeux d'essai dans un fichier `seed.sql` ou un export `.sql` pour l'évaluation.

Conclusion (exemple à insérer)

Le projet de refonte du site Decramp a permis d'appliquer un cycle complet : analyse, conception (UML/MERISE), développement, tests et déploiement. Les principaux livrables comprennent la maquette Figma, le code source PHP, la base de données MySQL et la documentation (cahier des charges, MCD/MLD/MPD). Les points forts du projet sont la sécurisation (hash mdp, PDO, HTTPS), la conformité RGPD partielle (bannière cookies et politique), et l'optimisation responsive et SEO. Les améliorations futures envisagées concernent l'ajout de tests E2E, la mise en place d'un monitoring (logs et alertes) et un tableau de bord admin plus complet.

11. Technologies utilisées

- **Front-end :** HTML5, CSS3, JavaScript (vanilla)
- **Back-end :** PHP 8
- **Base de données :** MySQL
- **Outils :** VS Code, Figma, GitHub, XAMPP/WAMP
- **Versionning :** Git
- **Hébergement :** Serveur PHP/MySQL
- **Sécurité avancée :** reCAPTCHA, mot de passe hashé, PDO préparé

13. Contraintes et normes

- _ Compatibilité navigateurs : Chrome, Edge, Firefox, Safari
- _ Responsive : mobile-first
- _ Respect du RGPD
- _ Normes W3C (HTML/CSS)
- _ Optimisation SEO (balises meta, titres hiérarchisés, texte alternatif)
- _ reCAPTCHA sur formulaires sensibles (contact).

15. Planning prévisionnel

Étape	Détail	Durée estimée
1. Analyse et maquette	Moodboard + wireframes (Figma)	1 semaine
2. Intégration HTML/CSS	Structure + design responsive	1 semaine
3. Intégration JS	Animation étoiles + menu burger	3 jours
4. Développement PHP	Formulaires + commentaires	1 semaine
5. Tests et validation	Tests multi-navigateurs et mobile	3 jours
6. Mise en ligne	Configuration serveur + déploiement	1 jour

16. Livrables

- Maquette graphique (Figma)
- Code source complet (HTML, CSS, JS, PHP)
- Base de données MySQL (structure + données de test)
- Cahier des charges (ce document)

