



RAPPORT DE PROJET

MODULE PROGRAMMATION PHP ET FRAMEWORK

CRM - Gestion de Contacts

Réalisé par :

BENCHAHID BASMA

BOURZGUI HIBA

HAMMOUDA IMANE

Soutenu le : 08/01/2026

Encadrée par :

Dr. YOUNES NADIR

ANNEE UNIVERSITAIRE : 2025/2026

DEDICACE

Nous dédions ce travail à nos parents, qui nous ont toujours soutenus et encouragés dans notre parcours académique et professionnel. Leur confiance et leurs conseils ont été une source constante de motivation.

Nous tenons également à exprimer notre gratitude envers nos enseignants et encadrants, dont l'accompagnement et les orientations nous ont permis de mener à bien ce projet. Leur exigence et leur disponibilité ont contribué à enrichir nos compétences et à renforcer notre rigueur méthodologique.

Enfin, nous dédions ce rapport à nos collègues et amis, pour leur soutien moral et leurs encouragements tout au long de cette réalisation. Ce projet est le fruit d'un travail collectif, d'échanges constructifs et d'une volonté partagée de progresser.

REMERCIMENT

Avant de débiter ce rapport, je tiens à exprimer ma gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce mini-projet.

Je remercie tout particulièrement **Monsieur YOUNES NADIR**, pour son encadrement, ses conseils précieux et sa disponibilité tout au long de ce travail. Ses orientations m'ont permis de mieux structurer mes idées et de progresser dans la mise en œuvre du projet.

Mes remerciements vont également à **l'ensemble des enseignants de l'EMSI**, pour la qualité des cours dispensés et l'environnement d'apprentissage qu'ils contribuent à créer chaque jour. Grâce à leur dévouement, j'ai pu acquérir les bases nécessaires à la conception et à la réalisation de ce projet.

Enfin, je souhaite remercier chaleureusement ma famille et mes proches, pour leur soutien moral, leur patience, et leur confiance tout au long de ce travail. Leur présence et leurs encouragements ont joué un rôle essentiel dans la réussite de ce projet.

À toutes et à tous, je vous exprime ma profonde reconnaissance.

RÉSUMÉ

Le projet consiste en une application web de gestion des contacts développée en **PHP natif** et partiellement ré-implémentée avec le framework **Symfony**. Elle permet de centraliser et de suivre les informations des contacts, offrant ainsi une gestion efficace et sécurisée. L'application prend en charge toutes les opérations de gestion : consultation de la liste des contacts, ajout de nouveaux, modification des informations, suppression protégée par confirmation, envoi d'emails via MailHog et gestion des tags associés.

L'architecture du projet repose sur un modèle **PHP natif** pour la partie cœur fonctionnel, et sur **Symfony** pour le module de listing, recherche et fiche contact. Les données sont persistées dans une base MySQL avec des relations entre utilisateurs, contacts et tags, tandis que Twig est utilisé pour l'interface utilisateur. La sécurité est assurée par un système de rôles (ROLE_USER et ROLE_ADMIN) et un **owner-check obligatoire**, garantissant qu'un utilisateur ne peut accéder qu'à ses propres contacts.

Le design a été pensé pour être pratique et ergonomique, intégrant des formulaires de recherche multi-critères avec tri, un upload sécurisé de photos (jpg/png/webp, max 2MB, renommage unique), ainsi que des cookies de préférences pour mémoriser le dernier filtre utilisé. Les tests d'acceptation incluent la vérification du contrôle d'accès (403 en cas de violation), la validation des uploads, la visibilité des emails dans MailHog et la robustesse de la recherche face aux caractères spéciaux.

Ainsi, cette application illustre l'utilisation combinée de **PHP natif et Symfony** pour développer une solution web complète, fiable et performante, permettant une gestion fluide des contacts tout en offrant une interface intuitive et sécurisée pour les utilisateurs et administrateurs.

LISTE DES ABRÉVIATIONS

ABRÉVIATION	SIGNIFICATION
CRUD	Create, Read, Update, Delete (Créer, Lire, Mettre à jour, Supprimer)
MVC	Model-View-Controller (Modèle-Vue-Contrôleur)
ORM	Object-Relational Mapping (Mapping Objet-Relationnel)
DB	Database (Base de données)
UI	User Interface (Interface utilisateur)
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
Twig	Template engine de Symfony
HTTP	HyperText Transfer Protocol
API	Application Programming Interface 'A FAIRE'
GET	Méthode HTTP pour récupérer des données
POST	Méthode HTTP pour envoyer des données
ID	Identifier (Identifiant unique)
EXP	Expéditeur
DES	Destinataire

Tables des matières

Introduction générale	8
1. Contexte générale et cahier des charges.....	9
1.1 Présentation de la problématique	9
1.2 Objectifs du projet... ..	10
1.3 Solution Proposée	12
1.4 Fonctionnalités du site... ..	13
1.5 Analyse de besoin.....	14
3. Conception	16
3.1 Diagrammes.....	16
3.1.1 Diagramme CLasse.....	17
3.1.1 Diagramme USE Case... ..	18
3.1.2 MVC.....	21
3.2 Etude de technologie et environnement	25
3.3 Outils de gestion du projet.....	26
4. Réalisation	27
4.1 Technologies utilisées.....	27
4.2 Interfaces de site... ..	28
4.3 Captures d'écran et explications	29

4.3.1	Liste complète des contact	29
4.3.2	Barre de recherche.....	30
4.3.3	Ajouter contact.....	31
4.3.4	Modifier contact.....	32
4.3.5	Affichage d'information du contact.....	33
4.3.6	Envoyer un email.....	34
4.3.7	Suppression du contact	35
5.	Conclusion et perspectives.....	37

INTRODUCTION GÉNÉRALE

La gestion efficace des contacts constitue un enjeu majeur pour les entreprises souhaitant centraliser et organiser leurs relations avec leurs clients, prospects et partenaires. Avec la dispersion des informations (téléphones, fichiers, carnets personnels), il est devenu essentiel de disposer d'un système capable de regrouper, suivre et sécuriser toutes les données relatives aux contacts. C'est dans ce contexte que j'ai choisi de développer une application web de gestion des contacts (CRM).

Ce projet a été réalisé en utilisant **PHP natif** pour le cœur fonctionnel et une ré-implémentation partielle avec le framework **Symfony**, reconnu pour sa robustesse et sa flexibilité dans le développement d'applications web.

L'application permet de gérer facilement les informations des contacts, telles que le nom, l'email, le téléphone, la ville, l'entreprise et les notes, tout en offrant des fonctionnalités avancées comme l'association de tags, l'upload sécurisé de photos, la recherche multi-critères avec tri et l'envoi d'emails via MailHog. Elle propose également toutes les opérations essentielles : consultation, ajout, modification et suppression protégée par confirmation, tout en garantissant une interface conviviale et intuitive.

L'architecture adoptée repose sur un modèle **MVC simplifié en PHP natif** et sur le modèle **MVC complet de Symfony** pour la partie listing et fiche contact, tandis que les contrôleurs orchestrent la logique métier et les vues **Twig** offrent une interface moderne et responsive. La sécurité est assurée par un système de rôles (**ROLE_USER** et **ROLE_ADMIN**) et un **owner-check obligatoire**, garantissant qu'un utilisateur ne peut accéder qu'à ses propres contacts.

Ainsi, ce projet constitue une solution complète et performante pour la gestion des contacts, illustrant concrètement l'utilisation combinée de **PHP natif et Symfony** pour le développement d'applications web ergonomiques et

fonctionnelles, tout en répondant aux besoins de fiabilité, de sécurité et d'efficacité dans la gestion des relations d'entreprise.

1) CONTEXTE GÉNÉRAL ET CAHIER DES CHARGES

1-1) PRÉSENTATION DE LA PROBLÉMATIQUE

Dans le domaine de la relation client et du développement commercial, la gestion des contacts représente un défi majeur pour les entreprises. Avec l'augmentation constante des échanges et la diversité des interlocuteurs (clients, prospects, partenaires), les informations sont souvent dispersées entre carnets personnels, fichiers Excel ou notes isolées. Sans un système efficace, il devient difficile de centraliser et de suivre précisément chaque contact, de retrouver rapidement ses coordonnées, ou encore de mettre à jour les informations relatives aux interactions. Cette absence de centralisation peut entraîner des pertes de données, des doublons, une mauvaise organisation et, in fine, une baisse de la qualité de service et de la satisfaction client.

La problématique à laquelle répond ce projet consiste donc à concevoir une application capable de centraliser et de gérer toutes les informations relatives aux contacts. L'objectif est de permettre un suivi précis de chaque relation, de faciliter la mise à jour des données et d'offrir une interface simple et intuitive pour l'utilisateur. Il s'agit de proposer une solution qui permette non seulement la consultation des contacts existants, mais également leur ajout, leur modification et leur suppression, tout en garantissant une expérience utilisateur agréable et un fonctionnement fiable.

Ainsi, le projet vise à résoudre les difficultés liées à la gestion manuelle des carnets d'adresses et à automatiser les opérations courantes, afin de gagner en efficacité, en précision et en qualité dans la gestion des relations professionnelles.

1-2) OBJECTIFS DU PROJET

Le projet (CRM : Contact Réseaux Managment) a pour objectif de concevoir une application web permettant de gérer efficacement les contacts en centralisant toutes les informations importantes et en automatisant les opérations courantes. L'application doit offrir une interface conviviale et intuitive afin de faciliter l'utilisation pour toute personne chargée de la gestion des relations clients, prospects ou partenaires.

L'un des principaux objectifs est de permettre le suivi complet de chaque contact, en enregistrant des données précises telles que le nom, l'email, le téléphone, la ville, l'entreprise et les notes. Cela garantit une gestion fiable et une meilleure organisation des informations, tout en réduisant les risques d'erreurs liés à une gestion manuelle et dispersée.

Le projet vise également à fournir une interface moderne et ergonomique, intégrant des fonctionnalités pratiques comme la recherche multi-critères avec tri, l'association de tags en relation many-to-many, l'upload sécurisé de photos (jpg/png/webp, max 2MB) et la mémorisation des préférences via cookies. Ces éléments améliorent l'expérience utilisateur et rendent l'application agréable à utiliser. Parallèlement, l'utilisation de **PHP natif** pour le cœur fonctionnel et de **Symfony avec Doctrine ORM** pour la ré-implémentation partielle assure la fiabilité et la sécurité des données, avec des opérations de persistance, de mise à jour et de suppression sécurisées.

Enfin, l'application a pour objectif de faciliter le suivi et le contrôle des contacts, offrant une vision claire de l'ensemble des relations et permettant une prise de décision rapide et efficace. L'ensemble de ces fonctionnalités vise à proposer une solution complète, performante et adaptée aux besoins modernes de gestion des contacts dans une PME.

1-3) PROBLEMATIQUE

Dans le contexte actuel, les petites et moyennes entreprises (PME) rencontrent des difficultés dans la gestion efficace de leurs contacts. Les informations relatives aux clients, prospects ou partenaires sont souvent dispersées dans différents supports (fichiers Excel, carnets d'adresses, emails, notes personnelles), ce qui entraîne une perte de temps, un manque de fiabilité et une absence de centralisation. Cette dispersion complique le suivi des relations, limite la traçabilité des échanges et augmente le risque d'erreurs ou de doublons.

De plus, l'absence d'un système structuré rend difficile la mise à jour des données, la recherche rapide d'informations pertinentes et la sécurisation des accès. Les utilisateurs n'ont pas toujours la possibilité de filtrer, trier ou organiser leurs contacts selon des critères précis, ce qui nuit à la performance et à la réactivité de l'entreprise. Enfin, la gestion collaborative est limitée, car il n'existe pas de mécanisme clair pour différencier les droits d'accès entre simples utilisateurs et administrateurs.

Ainsi, la problématique principale est la suivante : **comment concevoir une application web centralisée, fiable et ergonomique, permettant au\ PME de gérer efficacement leurs contacts, d'assurer la cohérence et la sécurité des données, tout en offrant une interface moderne et intuitive adaptée au\ besoins quotidiens des utilisateurs ?**

1-4) SOLUTIONS PROPOSÉE

Pour répondre à la problématique de gestion des contacts, la solution proposée consiste en le développement d'une application web dédiée, réalisée en **PHP natif** pour le cœur fonctionnel et partiellement ré-implémentée avec le framework **Symfony**. Cette application permet de centraliser toutes les informations relatives aux contacts au sein d'une base de données unique, garantissant ainsi une meilleure organisation et un accès rapide aux données. L'utilisation de Symfony offre une structure robuste, modulaire et adaptée au développement d'applications web modernes.

La solution repose sur l'architecture **MVC (Model-View-Controller)**, qui assure une séparation claire entre la logique métier, la gestion des données et l'interface utilisateur. Les données sont représentées par les entités **Users, Contacts et Tags**, ce qui permet une interaction fiable et sécurisée avec la base de données. Les contrôleurs assurent le traitement des requêtes, la validation des données et la communication entre le modèle et les vues, tandis que les templates Twig sont utilisés pour générer les interfaces utilisateur.

L'application permet une gestion complète des contacts à travers des fonctionnalités de consultation, d'ajout, de modification et de suppression protégée par confirmation. Elle intègre également des fonctionnalités

avancées telles que la recherche multi-critères avec tri, l'upload sécurisé de photos (jpg/png/webp, max 2MB, renommage unique), l'envoi d'emails via MailHog et la mémorisation des préférences grâce aux cookies. Chaque opération est réalisée via des formulaires interactifs, facilitant la saisie et la mise à jour des informations. Les données sont ensuite enregistrées ou mises à jour automatiquement dans la base de données, garantissant ainsi la cohérence et la fiabilité des informations.

Enfin, la solution intègre une interface utilisateur moderne et ergonomique, conçue pour améliorer l'expérience utilisateur. L'utilisation de composants visuels modernes, tels que des tableaux stylisés, des badges dynamiques indiquant les tags ou la ville, des animations légères et un mode sombre, permet de rendre l'application agréable à utiliser tout en restant fonctionnelle. Cette solution offre ainsi une réponse efficace, fiable et intuitive aux besoins de gestion des contacts dans une PME.

1-4) FONCTIONNALITÉS DU SITE

Le site développé offre une fonctionnalité principale de **consultation des contacts**, permettant à l'utilisateur d'afficher la liste complète des contacts enregistrés dans la base de données. Chaque contact est présenté avec ses informations essentielles, telles que le nom, l'email, le téléphone, la ville, l'entreprise et les tags associés. Cette vue globale permet un suivi clair et structuré de l'ensemble des relations de l'entreprise.

Le site permet également l'**ajout de nouveaux contacts** à travers un formulaire dédié. L'utilisateur peut saisir les informations nécessaires, notamment le nom, l'email, le téléphone, la ville, l'entreprise et les notes. Une photo peut être uploadée de manière sécurisée (jpg/png/webp, taille maximale 2MB, renommage unique). Une fois le formulaire validé, les données sont enregistrées dans la base de données et une page de confirmation s'affiche afin d'informer l'utilisateur du succès de l'opération.

Une autre fonctionnalité importante du site est la **modification des contacts existants**. À partir de la liste des contacts, l'utilisateur peut sélectionner un

contact spécifique et accéder à un formulaire prérempli avec ses informations actuelles. Cela permet de mettre à jour facilement les données, notamment en cas de changement d'adresse, d'entreprise ou de correction d'erreur lors de la saisie initiale.

Le site propose également la **suppression des contacts**. Cette fonctionnalité permet de retirer définitivement un contact de la base de données lorsque celui-ci n'est plus nécessaire. Elle est protégée par une confirmation afin d'éviter toute suppression accidentelle et contribue à maintenir une base de données propre et à jour.

Enfin, l'application propose une **interface utilisateur moderne**, avec un design responsive et un mode sombre afin d'améliorer le confort d'utilisation.

L'ensemble des fonctionnalités est accessible de manière intuitive, garantissant une navigation fluide .

1-5) ANALYSE DE BESOIN

Le projet de gestion des contacts répond à une problématique centrale : la nécessité pour une PME de disposer d'un outil fiable et centralisé permettant de gérer efficacement ses relations avec ses clients, prospects et partenaires. L'analyse des besoins met en évidence plusieurs axes essentiels :

1. Besoins fonctionnels

- **Consultation des contacts** : afficher la liste complète des contacts avec leurs informations principales (nom, email, téléphone, ville, entreprise, tags).
- **Ajout de nouveau\ contacts** : via un formulaire structuré permettant la saisie des données essentielles et l'upload sécurisé d'une photo.
- **Modification des contacts e\istants** : mise à jour des informations grâce à des formulaires préremplis.
- **Suppression des contacts** : suppression protégée par confirmation afin d'éviter les erreurs.
- **Recherche et filtrage** : recherche multi-critères (nom, ville, tags) et tri dynamique pour faciliter l'accès rapide aux données.

- **Envoi d'emails** : possibilité d'envoyer un message directement depuis la fiche contact, avec contrôle d'accès (owner-check).

2. Besoins techniques

- **Architecture MVC** : séparation claire entre la logique métier, la gestion des données et l'interface utilisateur.
- **Base de données relationnelle** : gestion des entités Users, Contacts et Tags avec Doctrine ORM pour assurer la fiabilité et la cohérence des données.
- **Sécurité** : owner-check pour restreindre l'accès aux données, confirmation avant suppression, contrôle des formats et tailles pour l'upload de photos.
- **Interface utilisateur** : utilisation de Twig, HTML, CSS et Bootstrap pour concevoir une interface moderne, responsive et ergonomique.

3. Besoins non fonctionnels

- **Ergonomie** : interface intuitive et accessible même pour des utilisateurs non techniques.
- **Performance** : rapidité d'accès aux données grâce à la centralisation et aux filtres.
- **Fiabilité** : cohérence des informations et traçabilité des modifications.
- **Accessibilité** : compatibilité multi-supports (ordinateur, tablette, smartphone).

4. Besoins organisationnels

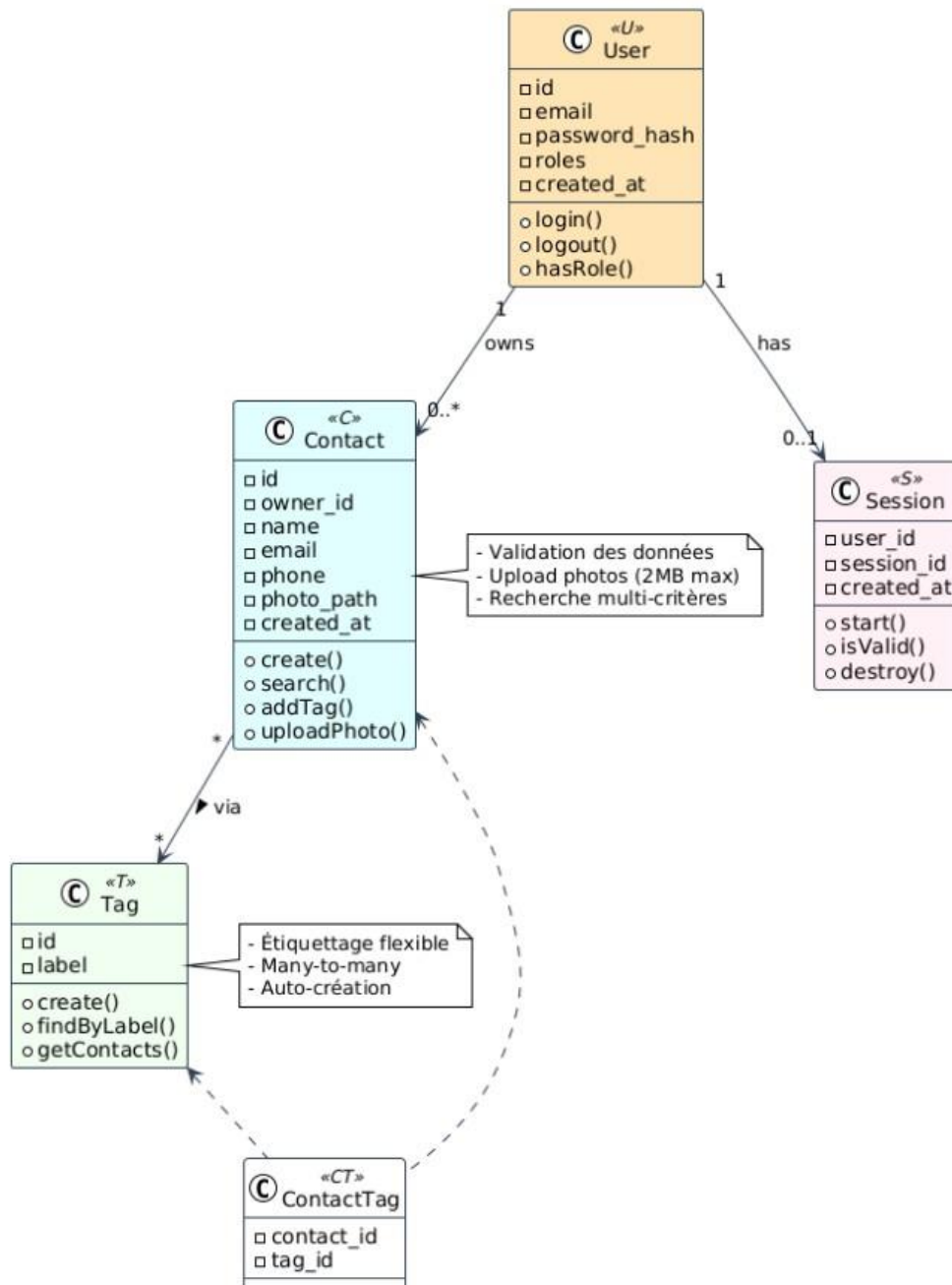
- **Centralisation des données** : éviter la dispersion des informations dans différents supports.
- **Collaboration** : permettre une gestion multi-utilisateurs avec rôles différenciés (utilisateur simple, administrateur).
- **Traçabilité** : conserver l'historique des créations et modifications pour un meilleur suivi.

3) CONCEPTION

3-1) LES DIAGRAMMES

3.1.1) diagramme de classe :

CRM HIBOS - Diagramme de Classes



RELATIONS EXPLIQUEES :

1. **User** 1----* **Contact** Un utilisateur peut avoir plusieurs contacts Chaque contact appartient    un seul utilisateur (owner_id)
2. **Contact** ---- **Tag** (via **ContactTag**) Un contact peut avoir plusieurs tags Un tag peut   tre associ      plusieurs contacts Relation many-to-many via table de liaison ContactTag

3. **User 1----**1 Session Un utilisateur peut avoir une session active Chaque session est liée à un utilisateur

METHODES PRINCIPALES :

User :

- **login()** : Authentification avec bcrypt
- **logout()** : Destruction de session
- **hasRole()** : Vérification rôle (ROLE_USER/ROLE_ADMIN)
- **getCurrentUser()** : Récupération utilisateur connecté

Contact :

- **create()** : Création avec validation
- **search()** : Recherche multi-critères
- **addTag()/removeTag()** : Gestion tags
- **uploadPhoto()** : Upload sécurisé (JPG/PNG/WEBP, max 2MB)

Tag :

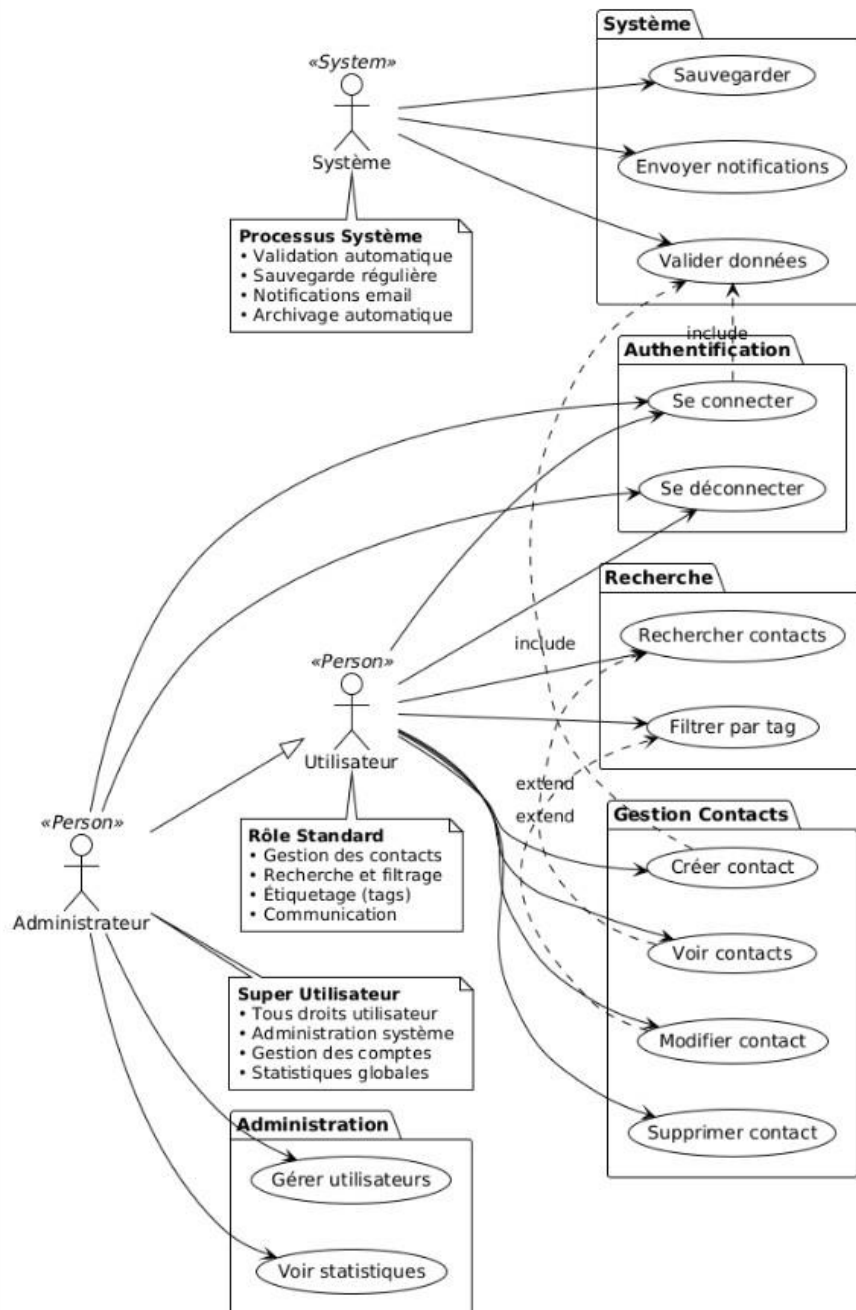
- **create()** : Création automatique si inexistant
- **findByLabel()** : Recherche par label
- **getContacts()** : Récupération contacts associés

Session :

- **start()** : Création session PHP
- **regenerate()** : Sécurité anti-CSRF
- **isValid()** : Vérification timeout

3.1.2) diagramme USE Case :

CRM HIBOS - Diagramme des Cas d'Utilisation



DESCRIPTION DES CAS D'UTILISATION PRINCIPAUX :

1. AUTHENTIFICATION

- **Se connecter** : Email + mot de passe

- **Se déconnecter** : Destruction session PHP
- **Vérifier droits** : Rôles ROLE_USER/ROLE_ADMIN

2. GESTION DES CONTACTS

- **Créer contact** : Formulaire + validation + upload photo
- **Consulter contacts** : Liste avec pagination + filtres
- **Modifier contact** : Mise à jour + gestion tags
- **Supprimer contact** : Confirmation + archive
- **Voir détails** : Fiche complète + historique
- **Uploader photo** : JPG/PNG/WEBP max 2MB

3. RECHERCHE ET FILTRAGE

- **Rechercher contacts** : Multi-critères (nom, email, téléphone, notes)
- **Filtrer par ville** : Sélection ville spécifique
- **Filtrer par tag** : Étiquettes multiples
- **Trier résultats** : Par nom, ville, date création
- **Sauvegarder filtres** : Cookies 30 jours

4. GESTION DES TAGS

- **Ajouter tags** : Création automatique si inexistant
- **Supprimer tags** : Suppression si non utilisé
- **Lister tags** : Affichage avec badges cliquables

5. COMMUNICATION

- **Envoyer email** : Template personnalisé + SMTP
- **Consulter historique** : Logs des communications

6. ADMINISTRATION

- **Gérer utilisateurs** : CRUD + rôles
- **Voir statistiques** : Dashboard avec graphiques
- **Exporter données** : CSV/PDF avec filtres

RELATIONS ENTRE CAS D'UTILISATION :

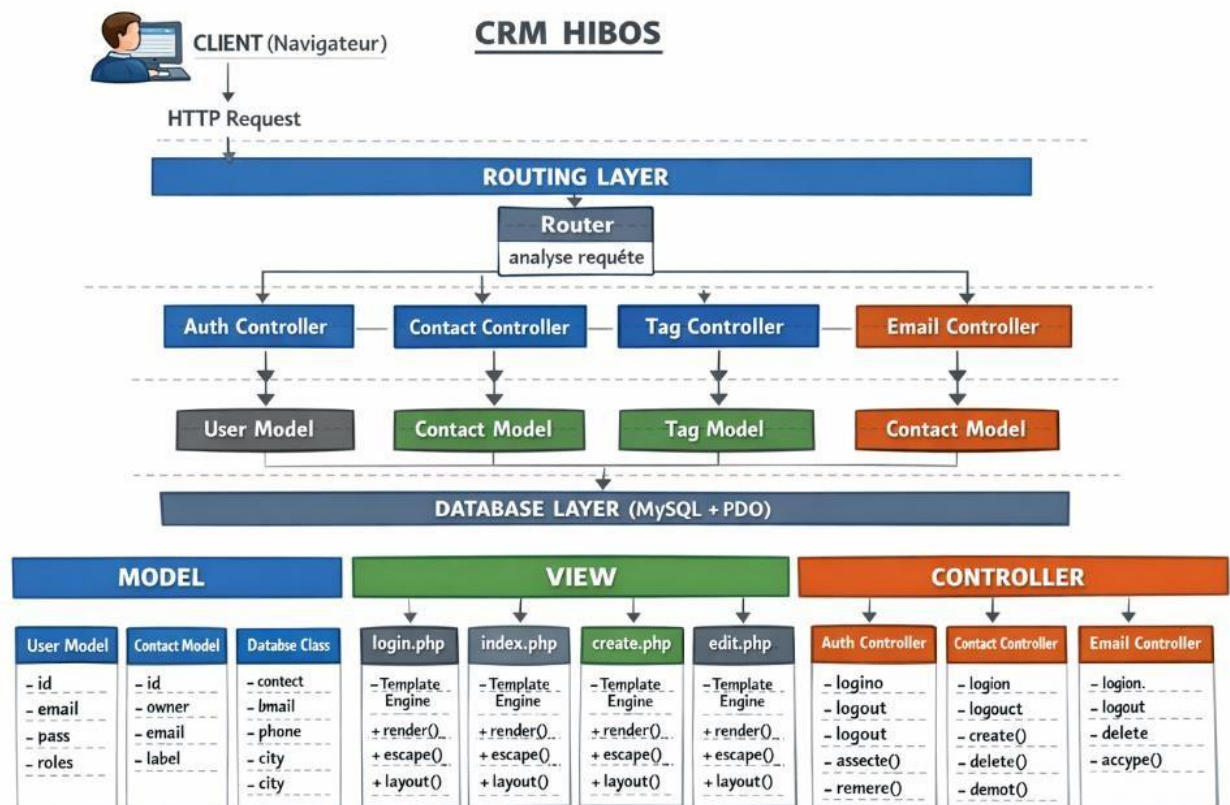
include (obligatoire) :

- **Se connecter** --> Vérifier droits
- **Créer contact** --> Ajouter tags
- **Modifier contact** --> Ajouter tags

extend (optionnel) :

- **Consulter contacts** --> Rechercher contacts
- **Consulter contacts** --> Filtrer par ville
- **Consulter contacts** --> Filtrer par tag
- **Voir détails** --> Envoyer email

3.1.3) MVC :



FLUX MVC COMPLET :

1. REQUEST

Client --> Router : HTTP/HTTPS Request

Router analyse : method, URI, params

2. ROUTING

Router --> Controller : Dispatch vers bon contrôleur

Router --> Middleware : Authentification, permissions

3. CONTROLLER

Controller --> Model : Appelle méthodes métier

Controller valide : Paramètres, permissions

4. MODEL

Model --> Database : Requêtes SQL

Model --> Database : Transactions

Database --> Model : Retourne données

5. RESPONSE

Model --> Controller : Données traitées

Controller --> View : Passe données

View --> Controller : HTML généré

Controller --> Client : HTTP Response

COMPOSANTS TECHNIQUES :

Routing Layer :

- **Router** : Analyse URL, dispatch
- **Request** : Wrapper HTTP (\$_GET, \$_POST, \$_FILES)
- **Response** : Génération réponse (redirect, json, view)

Model Layer :

- **Models** : Logique métier, validation
- **Database** : Connexion PDO, prepared statements
- **Transactions** : ACID, rollback/commit

View Layer :

- **Templates** : PHP avec HTML/CSS
- **Template Engine** : Échappement, layouts
- **Form Helper** : Génération formulaires sécurisés

Controller Layer :

- **Controllers** : Orchestration flux
- **Services** : Logique réutilisable
- **Middleware** : Auth, validation, logging

4) ÉTUDE DE LA TECHNIQUE ET DE L'ENVIRONNEMENT

1. PHP – Langage de programmation serveur

- **Historique** : PHP a été créé en 1994 par Rasmus Lerdorf sous le nom de *Personal Home Page Tools*. La première version publique date de 1995 . Initialement conçu pour suivre les visites sur son CV en ligne, PHP est

rapidement devenu un langage complet pour générer des pages web dynamiques.

- ☐ **Évolution** : Au fil des années, PHP a intégré des fonctionnalités avancées (programmation orientée objet, gestion des sessions, extensions multiples). Aujourd'hui, il est maintenu par la PHP Foundation et en est à la version 8.5 (2025).
- **Rôle dans le projet** : PHP natif constitue le cœur fonctionnel du CRM, permettant la gestion des opérations CRUD (ajout, modification, suppression, consultation) et l'interaction directe avec la base de données MySQL.

2. Symfony – Framework PHP




- **Historique** : Symfony a été créé en 2005 par Fabien Potencier au sein de l'agence française SensioLabs. La première version publique (0.4.0) est sortie en octobre 2005.
- ☐ **Évolution** : Symfony s'est imposé comme l'un des frameworks PHP les plus utilisés grâce à sa modularité et sa communauté active. Il repose sur l'architecture MVC et propose des composants réutilisables (Doctrine ORM, gestion des formulaires, sécurité). La dernière version stable est Symfony 8.0.2 (décembre 2025)
- **Rôle dans le projet** : Symfony a été utilisé pour réimplémenter certaines parties du CRM afin de bénéficier d'une structure robuste, d'une meilleure organisation du code et de fonctionnalités avancées comme la gestion des entités et la sécurité.

3. Bootstrap – Framework front-end

- **Historique** : Bootstrap a été développé en 2010 par Mark Otto et Jacob Thornton chez Twitter, sous le nom de *Twitter Blueprint*. Il a été publié officiellement en août 2011
- **Évolution** : Dès ses premières versions, Bootstrap s'est imposé comme le framework CSS le plus populaire grâce à ses composants prêts à l'emploi (grilles, formulaires, boutons, navigation). La version 5.3.8 (août 2025) est la plus récente
- **Rôle dans le projet** : Bootstrap a été utilisé pour concevoir une interface moderne, responsive et ergonomique. Il facilite la mise en page, l'adaptation aux différents écrans et l'intégration de styles cohérents (mode sombre, badges colorés, formulaires stylisés)...

3-3) OUTILS DE GESTION DU PROJET

Dans le cadre du développement de l'application CRM, plusieurs outils ont été mobilisés afin d'assurer une organisation efficace, une gestion optimale du code et une manipulation structurée des données :

-  **Bloc-notes** : utilisé pour la prise de notes rapides, la rédaction des idées initiales et la préparation des cahiers de charges ou des schémas de conception. Il a servi de support simple pour organiser les étapes du projet avant leur implémentation technique.
-  **Git** : outil de gestion de versions indispensable pour suivre l'évolution du code source. Il a permis de sauvegarder les différentes étapes du développement, de gérer les branches et de faciliter la collaboration en assurant la traçabilité et la sécurité des modifications.
-  **phpMyAdmin** : interface graphique utilisée pour administrer la base de données MySQL. Elle a facilité la création des tables (Users, Contacts, Tags), la gestion des relations, l'exécution de requêtes SQL et la vérification de la cohérence des données.



- **Visual Studio Code (VS Code)** : environnement de développement intégré (IDE) utilisé pour écrire, organiser et tester le code. Grâce à ses extensions (Symfony, PHP, Twig, Bootstrap), il a offert un cadre ergonomique et performant pour le développement, tout en améliorant la productivité et la lisibilité du projet.

4) RÉALISATION

4-1) TECHNOLOGIES UTILISÉES

Le développement de ce projet repose sur l'utilisation de plusieurs technologies modernes, choisies pour leur complémentarité et leur efficacité. La partie **cœur fonctionnel** a été réalisée en



PHP natif, permettant de mettre en place les modules essentiels (authentification, CRUD des contacts et tags, recherche multi-critères, upload sécurisé de photos, envoi d'emails via MailHog). Cette approche garantit une compréhension fine des mécanismes de base et une maîtrise des opérations fondamentales.

La ré-implémentation partielle a été effectuée avec



Symfony

le **framework Symfony**, reconnu pour sa robustesse et sa flexibilité dans le développement d'applications web. Symfony offre une architecture claire, favorisant une organisation structurée du code, une bonne maintenabilité et une sécurité renforcée. La gestion des données qui facilite la communication entre l'application et la base de données

relationnelle en manipulant les données sous forme d'objets, tout en garantissant la fiabilité des opérations de persistance, de mise à jour et de suppression.

La base de données a été gérée grâce à



MySQL, administrée via



phpMyAdmin, tandis que le serveur de développement local a été assuré par



XAMPP, offrant un environnement complet avec PHP, MySQL et Apache.

Pour la génération des interfaces utilisateur, le moteur de templates **Twig** a été utilisé afin de séparer la logique métier de la présentation et de produire des pages dynamiques et structurées. Les technologies



HTML et **CSS**, associées au framework, ont permis de concevoir une interface moderne, responsive et ergonomique.

4-2) INTERFACES DU SITE

L'interface du site a été conçue dans une optique de simplicité, modernité et ergonomie, afin de faciliter la navigation et l'utilisation des fonctionnalités par l'utilisateur. La structure des pages repose sur un design responsive, garantissant un affichage adapté sur différents supports, tels que les ordinateurs, les tablettes ou les smartphones. Chaque page est organisée de manière claire, avec des sections distinctes pour la consultation, l'ajout, la modification et la suppression des contacts.

La page d'accueil présente la liste complète des contacts sous forme de tableau, permettant à l'utilisateur de visualiser rapidement les informations essentielles de

chaque relation, telles que le nom, l'email, le téléphone, la ville, l'entreprise et les tags associés. Des actions directes, telles que la modification, la suppression ou l'envoi d'un email, sont accessibles via des liens intégrés dans le tableau, rendant la gestion plus intuitive et rapide.

Le site propose également des formulaires interactifs pour l'ajout et la modification des contacts. Ces formulaires sont préremplis lorsque nécessaire et

comportent des champs clairs et précis pour chaque donnée. L'upload sécurisé de photos est intégré, permettant d'associer une image à chaque contact. L'utilisation de badges colorés pour les tags, d'animations légères et d'un mode

sombre améliore l'expérience utilisateur, en rendant l'interface dynamique et agréable à manipuler. Le choix des couleurs, des polices et des éléments visuels a été fait dans le but de créer un design moderne tout en conservant une grande lisibilité et une ergonomie optimale.

Enfin, l'interface du site a été pensée pour être accessible et fonctionnelle, même pour un utilisateur non technique. Chaque fonctionnalité est facilement identifiable et accessible en quelques clics, permettant ainsi une gestion fluide des contacts et un suivi efficace des informations. L'ensemble de ces éléments

contribue à offrir une interface conviviale, moderne et adaptée aux besoins d'un système de gestion de contacts complet

4-3) CAPTURES D'ÉCRAN ET EXPLICATIONS

4-3-1) LISTE COMPLETE DES CONTACTS

NOM	EMAIL	TÉLÉPHONE	VILLE	ENTREPRISE	TAGS	ACTIONS
Basma.Benchahid	Basma.Benchahid@g.ma	06*****88	casa	hibos	--	<div>Voir</div> <div>Éditer</div> <div>Email</div> <div>Supprimer</div>
Hibaabourzgui	hibaabourzgui@gmail.com	06*****52	agadir	hibos	--	<div>Voir</div> <div>Éditer</div> <div>Email</div> <div>Supprimer</div>

La page **Liste des contacts** constitue la page principale de l'application et permet de visualiser tous les contacts enregistrés dans le système. Elle présente les informations essentielles de chaque contact sous forme de tableau clair et structuré, incluant le nom, l'adresse email, le numéro de téléphone, la ville, l'entreprise et les tags associés. Cette organisation permet à l'utilisateur d'avoir une vue globale et rapide de l'ensemble des relations, facilitant ainsi le suivi et la gestion des données.

Chaque ligne du tableau comporte des actions directes telles que consulter la fiche, modifier, envoyer un email ou supprimer un contact, permettant à l'utilisateur de gérer ses informations sans naviguer vers des pages complexes. Les colonnes sont disposées de manière logique pour rendre la lecture intuitive et réduire le temps nécessaire pour trouver une information précise. En résumé, la page Liste des contacts est conçue pour être fonctionnelle, intuitive et esthétique, permettant une gestion efficace et rapide des informations, tout en offrant une interface agréable et facile à utiliser.

4-3-2) BARRE DE RECHERCHE

RECHERCHER <input type="text" value="Nom, email, téléphone..."/>	VILLE <input type="text" value="Filtrer par ville"/>	TAG <input type="text" value="-- Tous les tags --"/>
TRI PAR <input type="text" value="Nom (A-Z)"/>	ORDRE <input type="text" value="Croissant (↑)"/>	<input type="button" value="RECHERCHER"/>

La page de recherche et de filtrage des contacts constitue une fonctionnalité essentielle du CRM, permettant à l'utilisateur de retrouver rapidement un contact spécifique parmi l'ensemble des données enregistrées. Elle est conçue pour être intuitive et complète, avec plusieurs champs de saisie et menus déroulants facilitant la recherche multi-critères.

L'utilisateur peut effectuer une recherche libre à partir du nom, de l'email ou du numéro de téléphone, filtrer les résultats par ville, sélectionner un tag spécifique pour affiner les résultats, et choisir un critère de tri (par nom, ville, entreprise, etc.) ainsi que l'ordre d'affichage (croissant ou décroissant). Ces options répondent aux exigences du cahier des charges en matière de recherche avancée, avec prise en charge des caractères spéciaux et mémorisation du dernier filtre utilisé via les cookies. L'interface est claire, responsive et agrémentée d'icônes visuelles pour chaque champ, rendant l'expérience utilisateur fluide et agréable.

En résumé, cette page permet une navigation rapide et efficace dans la base de contacts, tout en garantissant une organisation structurée et un accès ciblé aux informations pertinentes.

4-3-3) AJOUT DES CONTACTS



The screenshot displays the 'Ajouter un Contact' (Add Contact) form within the HIBOS CRM application. The interface includes a top navigation bar with the HIBOS CRM logo, a 'Contacts' menu item, an 'Ajouter' (Add) button, and a 'Déconnexion' (Logout) button. The main form area is titled '+ Ajouter un Contact' and contains a sub-header 'Remplissez les informations du nouveau contact'. The form fields are organized as follows: 'Nom Complet *' (required), 'Email', 'Téléphone', and 'Ville' are in the first row; 'Entreprise' and 'Notes' are in the second row; 'Photo' with a 'Choisir un fichier' button and 'Aucun fichier choisi' status is in the third row; and 'Tags (séparés par virgule)' with an example 'ex: VIP, Prospect, Client' and 'Tags existants' are in the fourth row. At the bottom of the form are two buttons: a blue 'AJOUTER' button and a light blue 'RETOUR' button.

La page **Ajouter un contact** constitue une étape clé dans le processus de gestion des données au sein du **CRM**. Elle permet à l'utilisateur de créer un nouveau contact en remplissant un formulaire structuré et intuitif. Ce formulaire comprend plusieurs champs essentiels tels que le nom complet, l'email, le numéro de téléphone, la ville, l'entreprise, les notes, ainsi qu'un champ pour l'upload sécurisé d'une photo (formats jpg, png, webp, taille maximale 2MB). L'utilisateur peut également associer des tags au contact, en les saisissant sous forme de liste séparée par des virgules, conformément à la relation many-to-many définie dans le modèle de données.

L'interface est conçue pour être claire et ergonomique, avec des libellés explicites et des exemples pour guider la saisie. Les boutons "AJOUTER" et "RETOUR" permettent respectivement de valider l'enregistrement ou de revenir à la liste des contacts. La navigation est fluide, et l'ensemble de la page respecte les principes du design responsive, garantissant une compatibilité optimale sur tous les types d'écrans.

En résumé, la page Ajouter un contact facilite la saisie rapide et fiable des informations, tout en respectant les contraintes techniques et les règles métier du projet. Elle contribue à enrichir la base de données du CRM de manière sécurisée et structurée, tout en offrant une expérience utilisateur agréable et efficace.

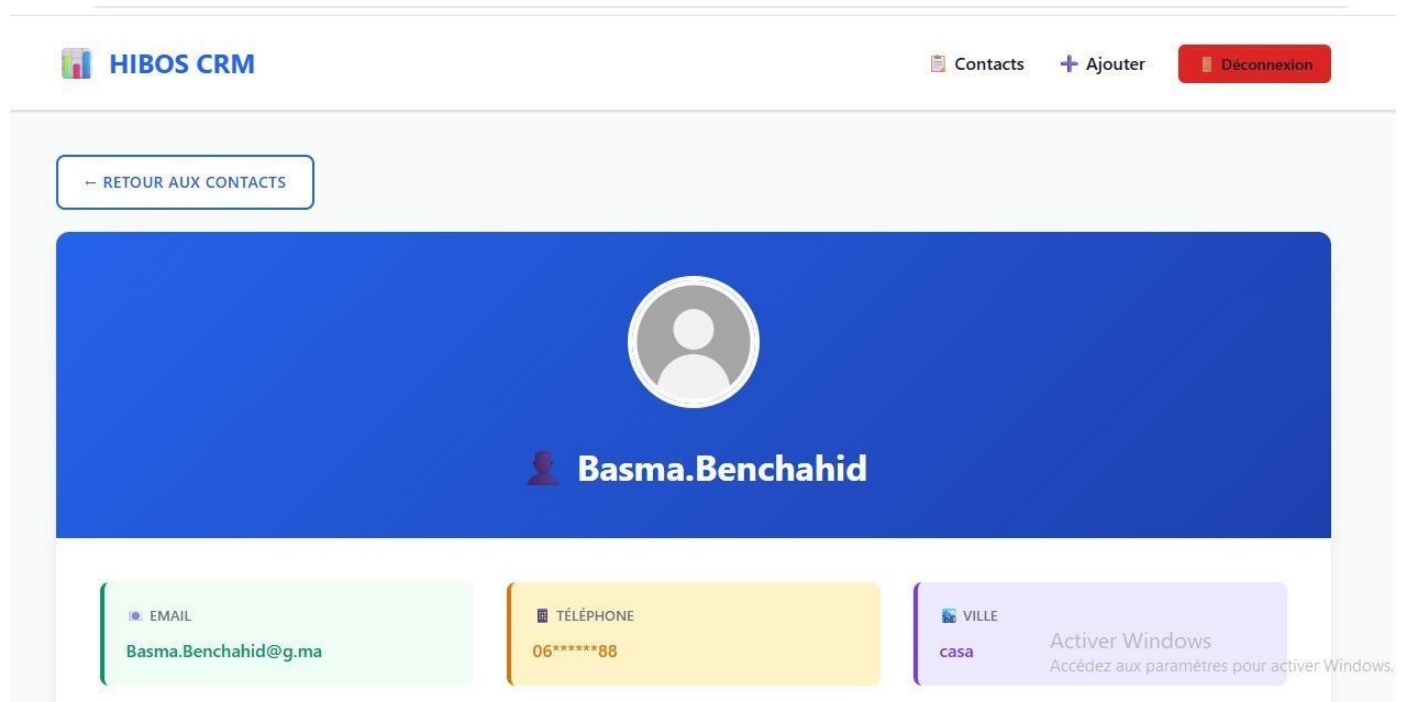
4-3-4) MODIFIER DES CONTACTS

La page **Modifier** un contact permet à l'utilisateur de mettre à jour les informations d'un contact existant dans le **CRM** de manière simple et sécurisée. Elle reprend les champs principaux du formulaire d'ajout, préremplis avec les données actuelles du contact : nom complet, email, téléphone, ville, entreprise, notes, photo et tags. L'utilisateur peut ainsi corriger ou compléter les informations sans avoir à ressaisir l'ensemble des données. Un champ dédié permet également de modifier ou ajouter des tags, en respectant la structure many-to-many définie dans le modèle de données.

L'interface est conçue pour être claire et intuitive, avec des boutons visibles pour valider les modifications ("METTRE À JOUR") ou annuler l'opération ("ANNULER"). L'upload de photo reste sécurisé, avec contrôle du format et de la taille, conformément aux règles métier du projet. La navigation est fluide et le design responsive garantit une compatibilité optimale sur tous les types d'écrans.

En résumé, la page Modifier un contact facilite la mise à jour rapide et fiable des données, tout en respectant les contraintes techniques et les règles d'accès du CRM. Elle contribue à maintenir une base de contacts propre, cohérente et à jour, tout en offrant une expérience utilisateur agréable et efficace.

4-3-5) AFFICHAGE D'INFORMATION DU CONTACT





La page **Fiche contact** permet d’afficher les détails complets d’un contact sélectionné dans le CRM. Elle présente les informations essentielles de manière structurée et lisible : le nom complet, l’adresse email, le numéro de téléphone, la ville, l’entreprise, ainsi que la photo de profil du contact. Ces données sont accompagnées de métadonnées comme l’identifiant unique du contact, la date de création et la date de dernière modification, ce qui facilite le suivi et la traçabilité des mises à jour.

L’interface propose également des actions directes accessibles depuis la fiche : modifier les informations du contact, envoyer un email via MailHog, ou supprimer le contact avec confirmation. Ces boutons sont clairement identifiés et permettent une gestion rapide sans quitter la page. Le design est épuré, responsive et pensé pour offrir une expérience utilisateur fluide, même sur mobile.

En résumé, la fiche contact constitue un point d’accès central pour consulter, gérer et interagir avec les données d’un contact, tout en respectant les règles d’accès et de sécurité définies dans le cahier des charges. Elle contribue à une gestion efficace, intuitive et sécurisée des relations professionnelles.

4-3-6) ENVOYER UN EMAIL

Envoyer un Email

À Contact (Basma.Benchahid@g.ma)

SUJET *

Objet de l'email

MESSAGE *

Votre message...

ENVOYER

RETOUR

Activer Windows

Accédez aux paramètres pour activer Windows.

La page **Envoyer** un email permet à l'utilisateur d'interagir directement avec un contact en lui adressant un message personnalisé via l'interface du CRM. Elle affiche clairement l'adresse email du destinataire, ici

« **Basma.Benchahid@g.ma** », et propose deux champs de saisie : le sujet de l'email et le corps du message. Ces champs sont obligatoires et conçus pour une saisie simple et intuitive, garantissant que l'utilisateur peut rédiger rapidement une communication professionnelle ou personnelle.

L'interface est épurée et ergonomique, avec des boutons bien identifiés pour valider l'envoi (« ENVOYER ») ou revenir à la page précédente (« RETOUR »). Cette fonctionnalité respecte les règles métier du projet, notamment le contrôle d'accès : seuls les utilisateurs autorisés (propriétaires du contact ou administrateurs) peuvent envoyer un email.

L'envoi est simulé via MailHog, permettant de vérifier la bonne transmission du message sans passer par un serveur réel.

En résumé, la page d'envoi d'email enrichit le CRM en offrant une fonctionnalité de communication directe, sécurisée et intégrée, tout en maintenant une interface claire et accessible pour tous les utilisateurs

4-3-7) SUPPRIMER CONTACT



Supprimer Contact

Êtes-vous sûr de vouloir supprimer ce contact ?

OUI, SUPPRIMER

NON, ANNULER

La page **Suppression** d'un contact constitue une étape critique dans la gestion des données du CRM, en garantissant que toute suppression est volontaire et confirmée par l'utilisateur. Lorsqu'un utilisateur choisit de supprimer un contact, une interface de confirmation s'affiche, lui demandant explicitement s'il souhaite poursuivre l'opération. Le message « Êtes-vous sûr de vouloir supprimer ce contact ? » est accompagné de deux boutons clairs : « OUI, SUPPRIMER » pour valider l'action, et « NON, ANNULER » pour revenir en arrière sans modifier la base de données.

Cette fonctionnalité respecte les règles métier du projet, notamment la protection contre les suppressions accidentelles. Elle est accessible uniquement aux utilisateurs autorisés (propriétaires du contact ou administrateurs), conformément au contrôle d'accès défini dans le cahier des charges. L'interface est sobre, intuitive et responsive, avec des éléments visuels comme l'icône de corbeille et des couleurs différenciées pour les actions critiques, renforçant la lisibilité et la sécurité de l'opération.

En résumé, la page de suppression assure une gestion rigoureuse et sécurisée des contacts, tout en offrant à l'utilisateur une interface claire et rassurante pour confirmer ses décisions.

5) CONCLUSION ET PERSPECTIVES

Le projet de gestion des contacts a permis de concevoir une application web complète et fonctionnelle, répondant aux besoins de consultation, d'ajout, de modification et de suppression des contacts. Grâce à l'utilisation de **PHP natif**, du framework **Symfony**, de **phpMyAdmin**, ainsi que de technologies web modernes telles que **HTML**, **CSS** et **Twig**, l'application offre une interface intuitive, ergonomique et responsive, facilitant la gestion efficace des informations. L'intégration de fonctionnalités visuelles comme le mode sombre, les badges colorés et les formulaires structurés a contribué à rendre l'expérience utilisateur agréable tout en garantissant la fiabilité et la sécurité des données.

Ce projet m'a permis de mettre en pratique les concepts fondamentaux du développement web, notamment l'architecture **MVC**, la gestion de bases de données relationnelles, la sécurisation des opérations **CRUD** et la création d'interfaces utilisateur claires et accessibles. Il a également illustré l'importance d'une interface bien pensée pour améliorer l'efficacité et la satisfaction de l'utilisateur final.

Pour l'avenir, plusieurs améliorations peuvent être envisagées afin d'enrichir et d'optimiser l'application. Parmi celles-ci, on peut citer l'implémentation d'un **tableau de bord statistique** pour analyser les contacts et leurs tags, l'ajout d'un **système de notifications internes** pour signaler les mises à jour ou les relances, ainsi que l'intégration d'une **gestion avancée des rôles et permissions** pour renforcer la sécurité et

favoriser un usage multi-utilisateurs. L'application pourrait également évoluer vers une solution collaborative, adaptée aux besoins des PME, en offrant une centralisation encore plus poussée des données et une meilleure traçabilité des interactions.