

RÉPUBLIQUE DU CAMEROUN
REPUBLIC OF CAMEROON
Peace – Work – Fatherland
UNIVERSITÉ DE DSCHANG
UNIVERSITY OF DSCHANG
Scholae Thesaurus Dschangensis Ibi Cordum
P 96, Dschang (Cameroun) -Tél. /Fax
(237) 233 45 13 811
Website: <http://www.univ-dschang.org>



FACULTE DES SCIENCES
FACULTY OF SCIENCE
Département de Mathématiques
et Informatique
Department of Mathematics and
Computer Science
BP 67, Dschang (Cameroun)
E-mail : udsrectorat@univ-dschang.org

RAPPORT DE FIN D'ETUDES EN CYCLE LICENCE EN INFORMATIQUE FONDAMENTALE

THEME : Création D'une Plateforme Dédiée Aux Photographes

Noms des participants

SOUS-GROUPES	NOMS ET PRENOMS	MATRICULES
FRONTEND-ANGULAR	LOMOFOUET NDONGMO HANDY JAURESS	CM-UDS-21SCI0124
	SOPIE POKAA MOREL	CM-UDS-21SCI0811
	NDE TIOMELA JORDAN	CM-UDS-20SCI1096
FRONTEND-IONIC	KOUOKAM TALLA EUGENE ASAPH	CM-UDS-21SCI0021
	METIEKAM KOAGNE MANUELLE PATRICIA	CM-UDS-21SCI0940
	GOUAFFO NANFAH JOHANN BRANDON	CM-UDS-18SCI2132
BACKEND	KENFACK FONGANG VICTOR CYNTICHE	CM-UDS-21SCI0555
	YOUAKAM FRANCLAIN CARELLE	CM-UDS-21SCI1162
	EWANE MBOBA HERMAN	CM-UDS-21SCI0286

Sous l'encadrement de : **Dr .ZEKENG MAXIME**

Année académique : 2023-2024

Table des matières

DEDICACE	4
REMERCIEMENTS.....	5
INTRODUCTION GENERALE.....	6
Chapitre I : PRESENTATION DU CADRE DE TRAVAIL ET DU CAHIER DE CHARGES	8
I.1 - Introduction.....	8
I.2 - Présentation du logo	8
I.3 - Cadre général et cadre spécifique du projet	9
I.3.1 - Cadre général.....	9
I.3.1 - Cadre spécifique	9
I.4 - Cahier de charges	10
I.5 - Conclusion	10
Chapitre II : CONCEPTION ET ANALYSE DU SYSTÈME.....	11
II.1 - Introduction.....	11
II.2 - Spécification des besoins.....	11
II.2.1 - Besoins fonctionnels	11
II.2.2 - Besoins non fonctionnels	12
II.3 - Présentation et limites du système existant.....	13
II.3.1 - Présentation.....	13
II.4 - Méthodologie et approche adoptées	14
II.5 - Diagramme de cas d'utilisation.....	14
II.5.1 - Identification des acteurs et des cas d'utilisation.....	14
II.6 - Gestion des données	17
II.6.1 - Dictionnaire de données.....	17
II.6.2 - Modèle Conceptuel de Données (MCD).....	19
II.6.2 - Modèle Logique de Données (MLD).....	19
II.7 - Diagramme de classe.....	20

II.8 - Diagrammes de séquence.....	20
a-Diagramme séquence pour s'inscrire.....	21
b-Diagramme pour s'authentifier.....	21
c-Diagramme pour contacter un photographe.....	22
d-Diagramme pour modification de la donnée par le photographe	23
II.9 - Diagrammes d'état transition	23
II.9.1 - Diagramme d'état transition pour la photo	23
II.10 - Diagramme d'activités	24
II.10.1 - Diagramme d'activités de S'authentifier et gérer compte	24
II.10.3 - Diagramme d'activités de Créer un compte.....	24
II.10.4 - Diagramme d'activités de Gérer un compte	25
II.11 - Diagramme de déploiement.....	25
II.12 - Conclusion.....	26
<i>Chapitre III : IMPLEMENTATION TECHNIQUE DE PROGRAMMATION ET</i>	
<i>RESULTATS</i>	27
III.1 - Introduction	27
III.2 - Environnement technique (choix des outils)	27
III.2.1 - Présentation des technologies et Framework Utilisés	27
III.2.1.1 - HTML.....	27
III.2.1.2 - CSS	27
III.2.1.3 - Laravel	28
III.2.1.4 - Angular	28
III.2.1.5 - Spring Boot.....	28
III.2.1.6 - UML	29
III.2.1.7 - MERISE.....	29
III.2.2 - Environnement de travail	29
III.2.2.2 - VS Code.....	29
III.2.2.3 - Chrome	29
III.2.2.4 - Git/Git Hub	30
III.2.2.5 - Android Studio	30
III.2.2.6- PostgreSQL.....	30

III.2.2.7 - Postman	30
III.2.2.8- Serveurs de développement	31
III.2.2.9 - Systèmes d'exploitation (Windows 10 et Ubuntu).....	31
III.4 - Architecture de notre plateforme.....	31
III.5 - Présentation de quelques résultats clés et Algorithmes d'obtention des résultats	32
III.5.1 - Front end.....	32
III.5.1.1 - Interface d'accueil	33
III.5.1.2 - Interface pour inscription du photographe.....	33
III.5.1.3 - Interface pour envoyer le code de vérification	34
III.5.1.4 - Interface pour connexion	34
III.5.1.5 - Interface pour accueil photographe	35
III.5.1.6 - Interface de Dashboard du photographe	35
III.5.1.7 - Interface pour créer une catégorie	35
III.5.1.8 - Interface pour publier une photo dans une catégorie.....	36
III.5.1.9 - Interface pour signature de l'image	36
III.5.1.10- Interface d'administration.....	37
III.5.2 - Backend	37
III.5.3 - Communication entre Frontend et Backend.....	37
<i>Chapitre IV : APPORT, LEÇON, DIFFICULTÉS ET SOLUTIONS.....</i>	<i>39</i>
IV.1 - Introduction.....	39
IV.2 - Apports et leçons tirées du projet.....	39
IV.2.1 - Apport pour nous.....	39
IV.2.2 - Apport pour la communauté.....	40
IV.3 - Problèmes rencontrés et quelques solutions	41
<i>CONCLUSION GENERALE</i>	<i>42</i>

DEDICACE

Nous dédions ce travail à nos familles respectives pour leur Amour inconditionnelle, le soutien constant et la guidance ont été notre phare tout au long de ce voyage. Leur foi inébranlable en nous a été une source constante d'inspiration et de motivation. À nos frères et sœurs, pour leur encouragement constant qui nous ont permis de persévérer dans les études et leur foi en nos capacités. À nos chers amis, pour leur compréhension, leur patience et leur soutien indéfectible tout au long de ce processus. Merci à chacun d'entre vous pour votre précieuse contribution à notre réussite. Ce travail est le fruit de votre soutien mutuel et de votre solidarité.

REMERCIEMENTS

Nous tenons à exprimer notre profonde gratitude à notre encadreur, le Dr. ZEKENG MAXIME, pour son soutien constant, ses conseils précieux et sa patience tout au long de ce travail. Son expertise et sa compréhension ont grandement contribué et ont été une source d'inspiration constante. Nous sommes également reconnaissants envers le personnel de l'Université de Dschang. Le dévouement de l'Université à l'éducation et à la recherche est une véritable inspiration pour nous tous. Leur soutien et leur encouragement ont été essentiels à la réalisation de ce travail. Enfin, nous remercions nos collègues et amis pour leur soutien moral et intellectuel. Leur camaraderie a rendu cette expérience d'autant plus enrichissante. Leur soutien et leur encouragement ont été une source de motivation constante. Nous sommes profondément reconnaissants à tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce travail. Chaque contribution, grande ou petite, a joué un rôle crucial dans notre parcours.

INTRODUCTION GENERALE

La photographie est un art de capturer l'instant, de figer les émotions et de raconter une histoire. Dans un monde où les images ont un pouvoir immense, les photographes jouent un rôle essentiel en capturant la beauté, l'authenticité et la diversité qui nous entourent. C'est dans l'optique de promouvoir l'art photographique que nous avons décidé de mettre sur pieds une plateforme dédiée aux photographes qui voudraient faire découvrir leur art, offrir leur service aux personnes qui souhaiteraient immortaliser un événement (cérémonie mariage, baptême, deuil...) et de passionner les personnes qui veulent elles aussi se lancer dans la photographie.

L'objectif premier de notre plateforme est de permettre aux photographes de s'authentifier, de gérer leurs publications, d'offrir la possibilité aux utilisateurs de faire des recherches les photos ou les photographes, de prendre des rendez-vous, d'aimer certaines photos, de commenter les profils des photographes, d'acheter certaines images à leur disposition sur la plateforme grâce au système de paiement mis en place, de permettre aux administrateurs de s'assurer que les photos proposer par les photographes respectent la politique de la plateforme mais aussi de garantir la satisfaction des utilisateurs c'est-à-dire s'assurer que chaque photographe honore un contrat passer avec un client. La plateforme aura aussi pour but de d'offrir aux utilisateurs des services de qualité, permettre aux photographes d'exposer son savoir-faire en ce qui concerne la capture des images, la plateforme va permettre aux photographes de protéger leurs œuvres cela grâce au système de détection d'images postées de façon illégale sur d'autres plateforme, la plateforme va permettre aux photographes de signer leurs œuvres.

Notre plateforme a été conçue à l'aide de plusieurs technologies web notamment les Framework (Laravel pour la création des systèmes de routages, Angular qui utilise le langage TypeScript pour faciliter la création d'interfaces utilisateur réactives, Ionic pour créer une application mobile de la plateforme, string Boot pour faciliter la création de micro services et d'applications RESTful), nous avons utilisé les APIs (Application Programming Interface) pour la mise en place des systèmes bancaires, les applications de modélisations telles que Win'Design, Modélio et Star UML pour la conception des modèles (mcd,mld, diagrammes des cas d'utilisations, les diagrammes de séquences), les éditeurs de code tels que VS Code, Android Studio, un système de gestion de base de données tel que PostgreSQL.

Notre travail est reparti en quatre chapitres, dans le premier chapitre nous ferons la description du logo, la présentation des spécificités de la plateforme. Ensuite, nous présenterons dans le second chapitre les besoins fonctionnels, non fonctionnels, les limites existantes de la plateforme, l'ensemble des diagrammes utilisés pour la conception de la plateforme, le dictionnaire de données. Dans le troisième chapitre, nous listerons les outils utilisés pour la création de l'application. Et enfin dans le dernier chapitre, nous vous présenterons les différentes interfaces (Dashboard de l'administrateur, le Dashboard du photographe, la page d'accueil de la plateforme, l'interface de connexion, de vérification de mot de passe), l'architecture de la plateforme, les étapes de communication entre le front end et le back end, les problèmes rencontrés et les solutions apportées, les apports de la plateforme.

Chapitre I : PRESENTATION DU CADRE DE TRAVAIL ET DU CAHIER DE CHARGE

I.1 - Introduction

Dans un monde de plus en plus visuel et connecté, la photographie occupe une place prépondérante dans notre quotidien. Que ce soit pour immortaliser des moments précieux, mettre en valeur des produits ou encore partager des instants de vie sur les réseaux sociaux, la demande en services photographiques ne cesse de croître. C'est dans ce contexte que notre plateforme de photographes voit le jour, avec pour objectif de mettre en relation des talents de la photographie avec des clients en quête d'images exceptionnelles.

Notre plateforme se veut être un véritable carrefour de créativité et de l'expertise, offrant aux photographes la possibilité de mettre en avant leur talent et leur savoir-faire, tout en proposant aux clients une sélection diversifiée de professionnels qualifiés. Grâce à une interface intuitive et conviviale, nous facilitons la recherche et la réservation des photographes, tout en garantissant une expérience client optimale.

Que vous soyez à la recherche d'un photographe pour un événement spécial, une séance photo professionnelle ou tout simplement pour immortaliser des moments uniques, notre plateforme met à votre disposition un large choix de profils pour répondre à tous vos besoins. Nous mettons un point d'honneur à promouvoir la qualité et la diversité artistique, afin de satisfaire les attentes les plus exigeantes.

I.2 - Présentation du logo



Notre logo nommé « FISHAYE » Porte le nom de notre plateforme dédié aux photographes, il est stylisé en noir et blanc mettant en scène un photographe et une femme en profil. Le photographe, à gauche est capturé en pleine en pleine action avec son appareil photo, représentant la passion et l'art de la photographie. A droite une femme au profil élégant et aux cheveux afro, dégage une aura de sérénité et de dignité. Le nom « FISHAYE » en lettres capitales sobres et modernes est centré en dessous, soulignant le professionnalisme des photographes et la simplicité de la plateforme.

I.3 - Cadre général et cadre spécifique du projet

I.3.1 - Cadre général

Notre plateforme de photographes vise à répondre à la demande croissante en service de photographie, en mettant en relation des photographes professionnels avec des clients à la recherche des images de qualité. Nous souhaitons créer un espace de rencontre et d'échange entre les talents de photographie et les personnes désireuse de bénéficier de leurs compétences.

I.3.1 - Cadre spécifique

Notre plateforme propose une interface conviviale permettant aux clients de rechercher, comparer et réserver des photographes en fonction de leurs besoins spécifique. Les photographes peuvent créer leur profil, présenter leur portfolio et leurs services, tout en gérant facilement leurs disponibilités et leurs tarifs.

Les clients ont la possibilité de parcourir les profils des photographes, consulter leur réalisation passée, lire les avis des précédents clients à travers les commentaires et effectuer des réservations en ligne. Nous garantissons une expérience fluide et sécurisée, facilitant ainsi la mise en relation entre les deux parties.

Nous accordons une importance particulière à la qualité artistique des photographes présents sur notre plateforme, en veillant à promouvoir la diversité des styles et des approches créatives. Nous encourageons l'échange et la collaboration entre les membres de notre communauté, dans le but de favoriser l'épanouissement professionnel et artistique de chacun.

I.4 - Cahier de charges

Le cahier des charges de ce projet détaille les exigences spécifiques de la plateforme. Il comprend les fonctionnalités clés qui doivent être implémentées, ainsi que les attentes en termes de performance et de fiabilité. Les principales fonctionnalités de la plateforme comprennent la possibilité pour les utilisateurs de s'authentifier, de publier des offres, de gérer leurs comptes, de rechercher les photographes, de faire des réservations, liker et de commenter les profils des photographes. En outre, le cahier des charges stipule que la plateforme doit offrir une interface utilisateur intuitive, garantir la sécurité des données des utilisateurs, et être capable de gérer un grand nombre d'utilisateurs simultanément.

I.5 - Conclusion

En résumé, notre plateforme de photographes se positionne comme un lieu d'échange, de découverte et de valorisation du talent photographique, offrant aux clients un accès facilité à des services professionnels et personnalisés, tout en soutenant les photographes dans leur développement et leur visibilité sur le marché.

Chapitre II : CONCEPTION ET ANALYSE DU SYSTÈME

II.1 - Introduction

La conception et l'analyse du système sont des étapes cruciales dans le développement de toute application informatique. Elles permettent de comprendre comment les données seront stockées, manipulées et récupérées dans le système, comment les différentes fonctionnalités seront implémentées, et comment le système sera structuré et organisé. Dans ce chapitre, nous présenterons la conception et l'analyse que nous avons effectuées pour notre plateforme dédiée aux photographes. Nous discuterons des structures de données que nous avons choisies pour stocker les informations sur les utilisateurs, les photos, les réservations et les avis, nous présenterons les diagrammes UML que nous avons utilisés pour modéliser le système, et nous discuterons également de l'architecture de l'application et du modèle de conception que nous avons adopté.

II.2 - Spécification des besoins

II.2.1 - Besoins fonctionnels

Les besoins fonctionnels décrivent les fonctionnalités que le système doit offrir. Pour notre plateforme dédiée aux photographes, les besoins fonctionnels comprennent :

- **Authentification** : Les utilisateurs (photographes et administrateurs) doivent être en mesure de créer un compte, de se connecter et de se déconnecter de la plateforme.
- **Gestion des offres** : Les photographes doivent être en mesure de publier des photos, de les modifier et de les supprimer.
- **Recherche de photographes** : Les clients doivent être en mesure de rechercher des photographes en fonction de divers critères, tels que la localisation, leur noms, une petite description, la catégorie des photos.
- **Réservation des photographes** : Les clients doivent être en mesure de réserver des photographes directement sur la plateforme.

- **Likes et commentaires** : Les clients doivent être en mesure de liker et de commenter les profils des photographes avec qui ils ont travaillé.

II.2.2 - Besoins non fonctionnels

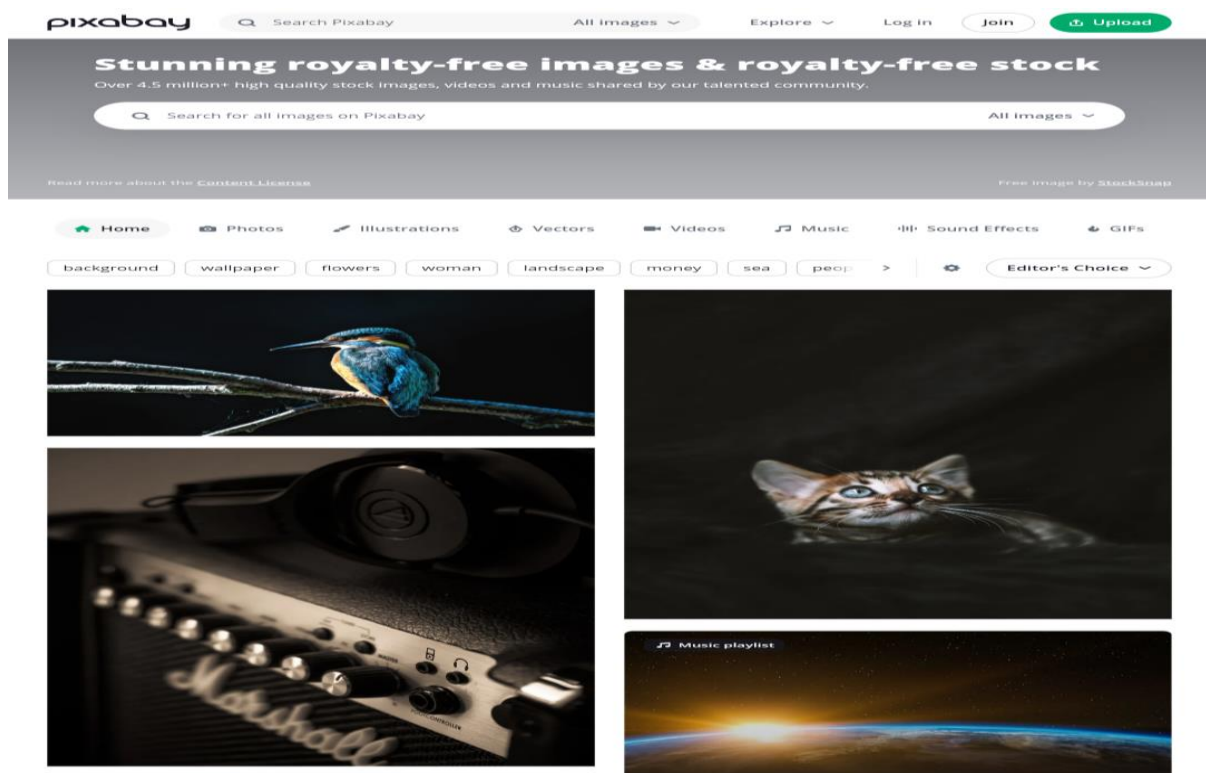
Les besoins non fonctionnels sont des caractéristiques du système qui décrivent comment il doit fonctionner, plutôt que ce qu'il doit faire. Voici quelques exemples de besoins non fonctionnels pertinents pour notre projet :

- **Performance** : Le site doit être rapide et réactif, même lorsqu'il est utilisé par de nombreux utilisateurs simultanément. Les temps de chargement des pages et des images doivent être minimaux.
- **Sécurité** : Le site doit garantir la sécurité des données des utilisateurs, y compris les informations personnelles des photographes et des clients.
- **Disponibilité** : Le site doit être disponible en tout temps, avec un minimum de temps d'arrêt planifié pour la maintenance. Il doit également être capable de gérer les pics de trafic sans compromettre la disponibilité.
- **Évolutivité** : Le site doit être conçu pour être évolutif, capable de gérer une croissance future du nombre d'utilisateurs, de photographes et de fonctionnalités sans nécessiter de refonte majeure.
- **Fiabilité** : Le site doit être fiable, avec un faible taux d'erreurs et de pannes. Les fonctionnalités critiques telles que la réservation des services photographiques doivent fonctionner de manière fiable à chaque fois.
- **Convivialité** : L'interface utilisateur doit être conviviale et intuitive, facilitant la navigation et l'utilisation du site pour les photographes et les clients.
- **Compatibilité** : Le site doit être compatible avec différents navigateurs web, appareils et systèmes d'exploitation pour assurer une expérience utilisateur cohérente.
- **Maintenabilité** : Le site doit être facile à maintenir, avec un code bien structuré et des fonctionnalités modulaires qui permettent des mises à jour et des améliorations faciles.
- **Performance SEO** : Le site doit être optimisé pour les moteurs de recherche (SEO) afin d'assurer une visibilité maximale dans les résultats de recherche.

II.3 - Présentation et limites du système existant

II.3.1 - Présentation

Le système existant que nous avons examiné est PIXABAY, une plateforme en ligne qui offre une vaste collection d'images, de vidéos, de vecteurs et d'illustrations libres de droits. Les utilisateurs peuvent accéder à de millions de ressources visuelles de haute qualité, téléchargeable gratuitement et utilisable à des fins personnels et commerciales, sans avoir à obtenir l'autorisation de l'auteur ou à payer des droits d'auteur. Les avantages de pixabay incluent la diversité des contenus proposés, la facilité d'utilisation de la plateforme, ainsi que la possibilité de télécharger et d'utiliser les ressources sans se soucier des droits d'auteur.



II.3.2 - Les limites

Malgré de nombreux avantages, pixabay présente également quelques limites. Tout d'abord, étant donné que les images sont partagées par des contributeurs individuels, la qualité et la pertinence des contenus peuvent varier. De plus, bien que les ressources soient généralement libres de droit, il est important de lire les licences associées à chaque image ou vidéo pour s'assurer qu'elles correspondent à l'utilisation prévue. En fin de compte, comme pour toute plateforme en ligne reposant sur des contributions communautaires, il est possible de rencontrer du contenu inapproprié ou de mauvaise qualité.

II.4 - Méthodologie et approche adoptées

Pour ce projet, nous avons adopté une approche itérative et incrémentale, en commençant par une analyse détaillée des besoins et une spécification claire des fonctionnalités requises. Nous avons ensuite conçu les différents diagrammes UML pour modéliser le système, y compris les diagrammes de cas d'utilisation, de classes et de séquences. Enfin, nous avons défini le dictionnaire de données et le modèle conceptuel de données (MCD) pour structurer les données du système.

II.5 - Diagramme de cas d'utilisation

II.5.1 - Identification des acteurs et des cas d'utilisation

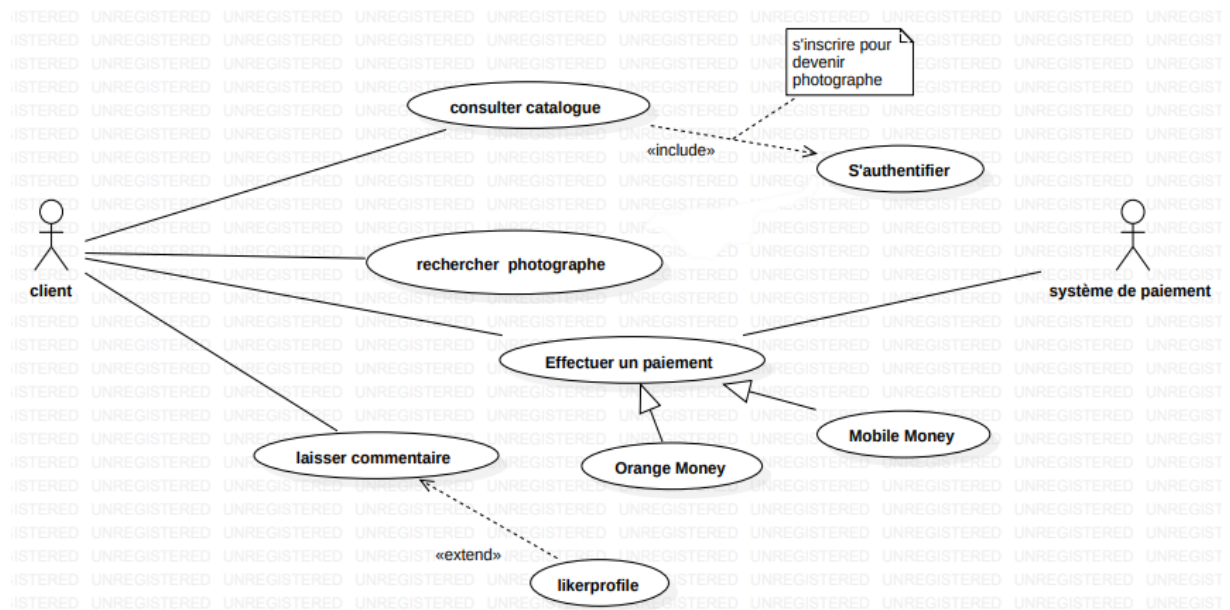
Dans notre système, nous avons identifié plusieurs acteurs qui interagissent avec le système de différentes manières. Ces acteurs sont les suivants :

- **photographe** : Il s'agit de l'acteur principal qui interagit avec le système. IL peut, créer un compte, s'authentifier, gérer son profil, gérer ses publications.
- **client**: C'est un utilisateur qui est à la recherche d'un photographe pour ses besoins spécifiques, il peut rechercher les photographes, prendre un rendez-vous, effectuer un contrat et un paiement.
- **Administrateur** : C'est l'administrateur du système. Il a la responsabilité de gérer l'ensemble du système, y compris les utilisateurs.

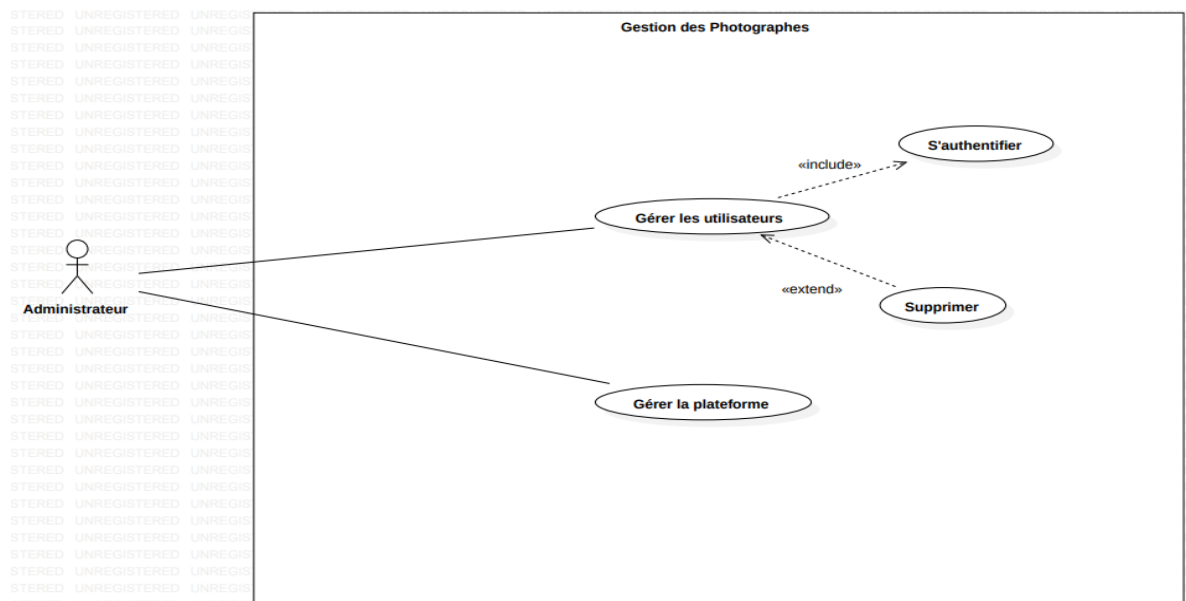
Les cas d'utilisation identifiés sont les suivants :

- Photographe et Administrateur (Créer un compte, s'authentifier, Gérer son profil)
- Administrateur (Gérer les utilisateurs et la plateforme)
- Photographe (Publier une photo, gérer les rendez-vous, consulter ses messages, consulter les commentaires)
- Client (Rechercher un photographe, réserver un photographe, effectuer un paiement, liker et commenter un profil, contacter un photographe)

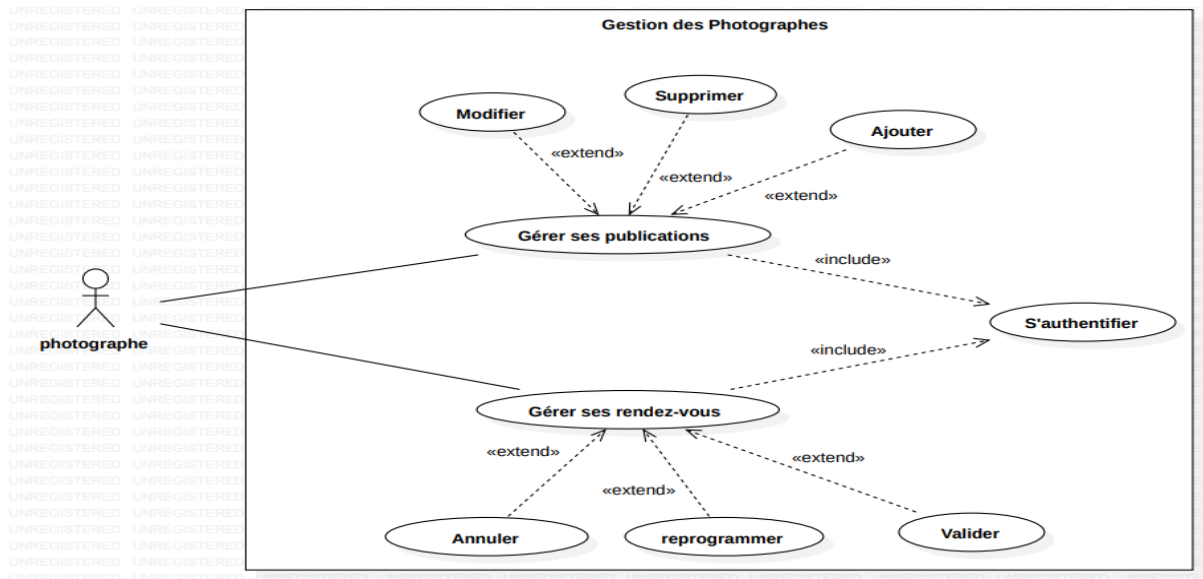
a-Diagramme de cas d'utilisation du client



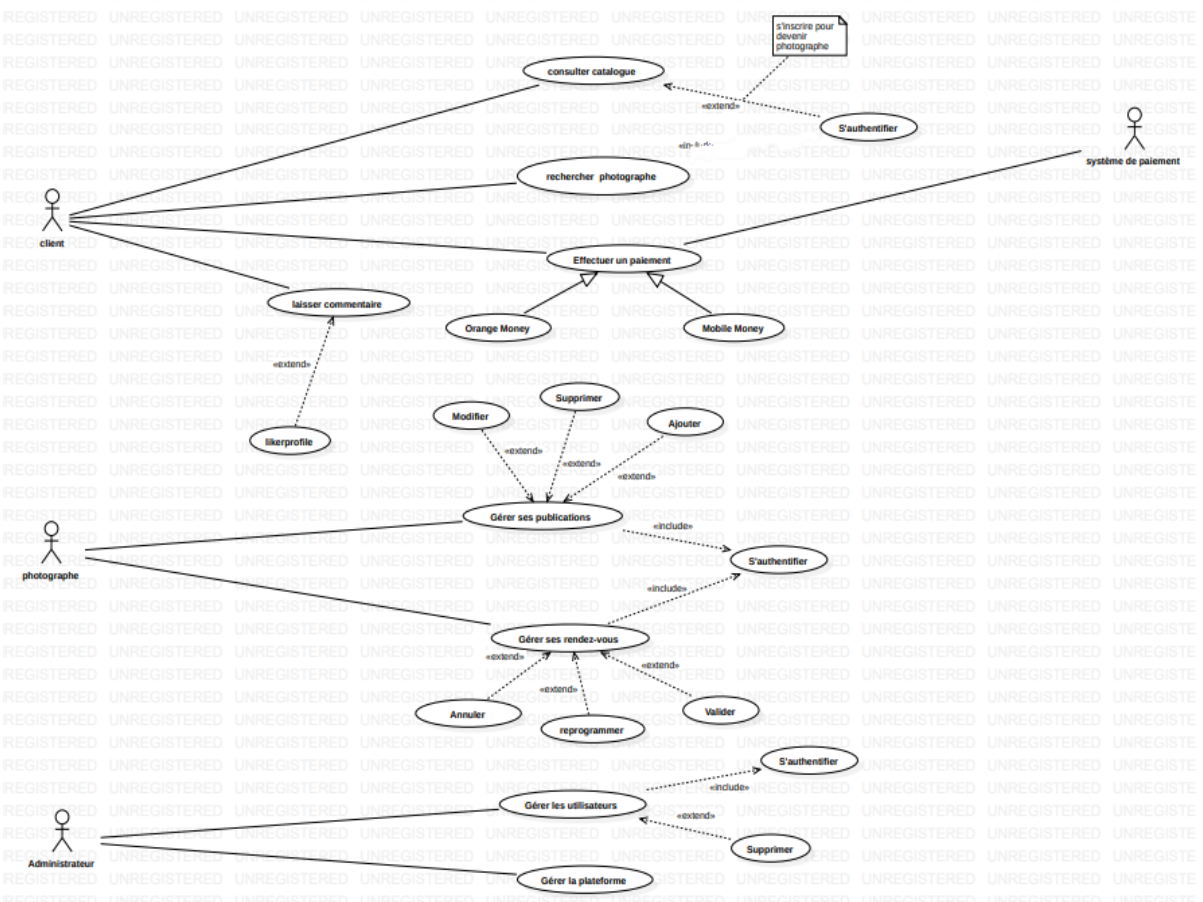
b-Diagramme de cas d'utilisation de l'administrateur



c-Diagramme de cas d'utilisation du photographe



d-Diagramme de cas d'utilisation globale



II.6 - Gestion des données

La gestion des données est une partie essentielle de notre système. Elle permet de structurer, stocker et manipuler les informations nécessaires au fonctionnement du système. Dans cette section, nous allons décrire le dictionnaire de données et le Modèle Conceptuel de Données (MCD) que nous avons élaborés pour notre système.

II.6.1 - Dictionnaire de données

Le dictionnaire de données est un document qui décrit les données utilisées dans le système. Il contient des informations sur chaque entité, y compris son nom, sa description, son type, sa longueur, sa nature et son format. Voici un extrait de notre dictionnaire de données :

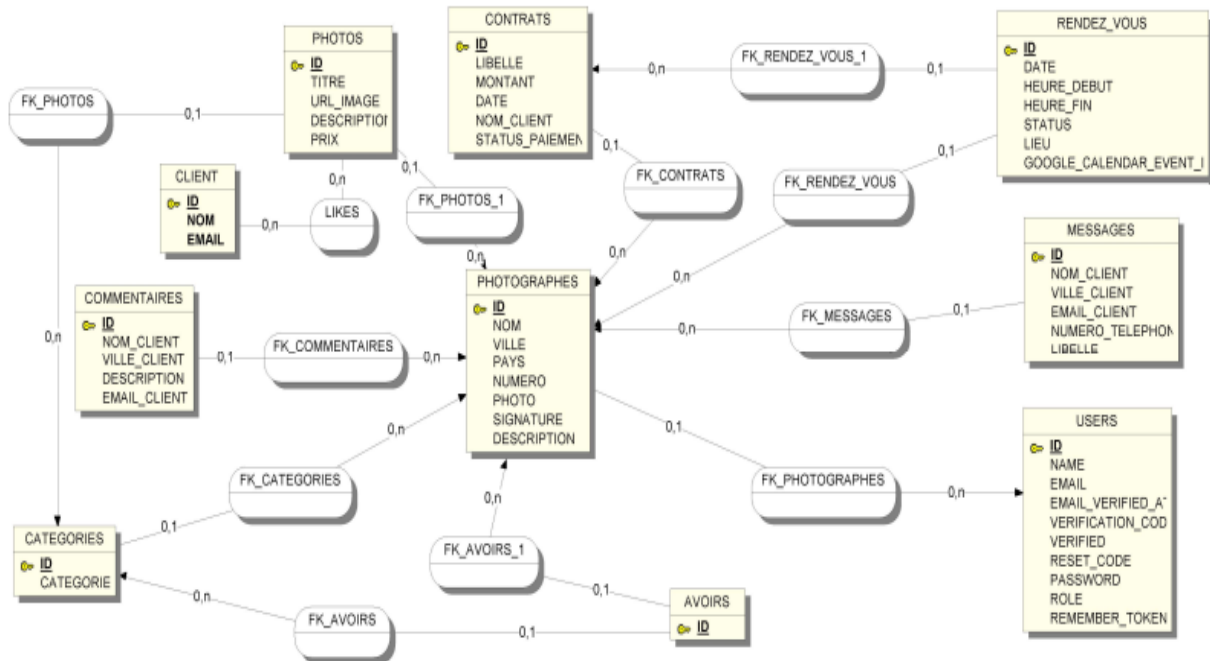
DICTIONNAIRE DE DONNÉES

Code	Signification	Type	Nature	Longueur	Observation
Photographe_id	Code du photographe	AN	obligatoire	5	Identifiant
Nom_Photographe	Nom du photographe	A	obligatoire	20	
pays_photographe	Pays du photographe	A	obligatoire	20	
Email_photographe	Email du photographe	AN	obligatoire	25	
ville_photographe	Ville du photographe	A	obligatoire	20	
mot_de_passe	Mot de passe du photographe	AN	obligatoire	20	Elle doit être masquée
numero_photographe	Numéro du photographe	N	obligatoire	12	
description_photographe	Description du photographe	TEXTE	obligatoire	256	
minPrix_par_heure	Prix minimal d'une séance par heure	AN	obligatoire	5	
categorie_id	Code d'une catégorie	AN	obligatoire	5	Identifiant
libellé	Libellé de la catégorie		obligatoire	20	
disponibilité_id	Code de la disponibilité	AN	obligatoire	5	identifiant
date	Date de disponibilité	DATE	obligatoire	10	JJ/MM/AAAA
heure	Heure de disponibilité	AN	obligatoire	6	
commentaire_id	Code d'un commentaire	AN	obligatoire	5	identifiant
nom_client	Nom du client qui émet un commentaire	A	obligatoire	20	
ville_client	Ville du client	A	obligatoire	20	
email_client	Email du client	AN	obligatoire	25	
numero_client	Numéro du	N	obligatoire	12	

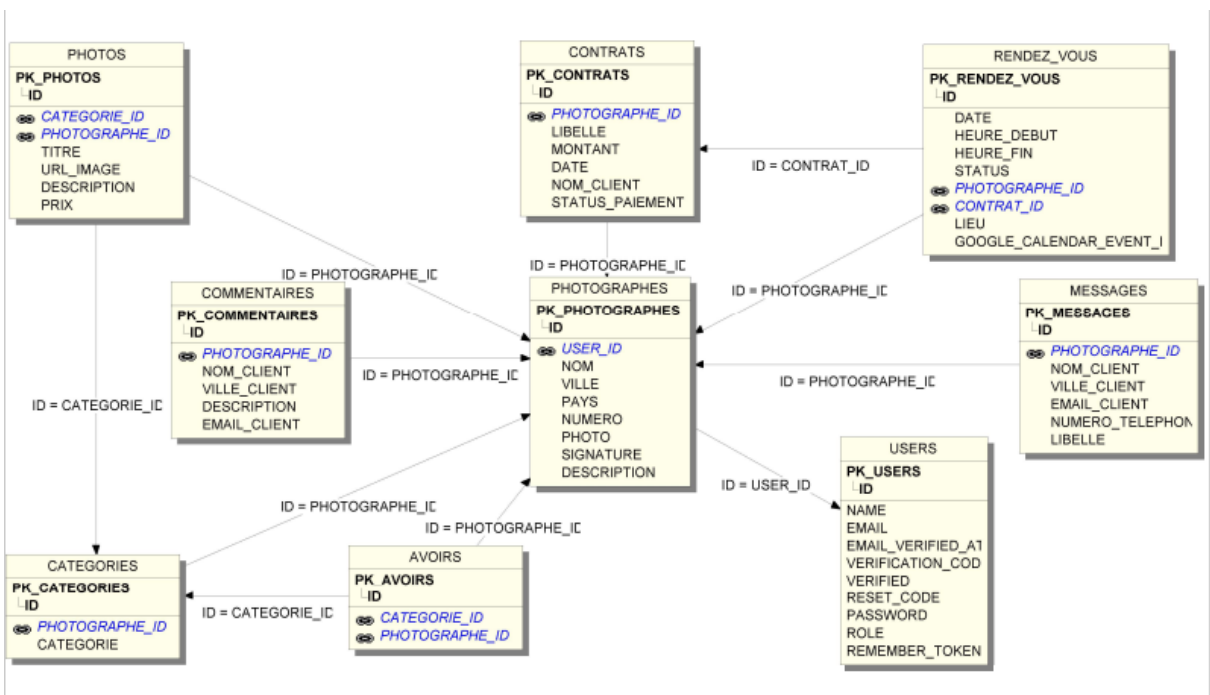
	client				
description	Intitulé du commentaire	TEXTE	obligatoire	256	
photo_id	Code de la photo	AN	obligatoire	5	identifiant
titre	Nom de la photo	A	obligatoire	20	
description_photo	Description de la photo	AN	obligatoire	256	
url_image	photo	AN	obligatoire	256	
nombre_like	Mention "j'aime" sur une photo	N	facultatif	5	
rendez_vous_id	Code du rdv	AN	obligatoire	5	identifiant
date	Date du rdv	DATE	obligatoire	10	JJ/MM/AAAA
heuredebut	Heur du début du rdv	AN	obligatoire	6	
heurefin	Heure de fin du rdv	AN	obligatoire	6	
lieu	Lieu du rdv	AN	obligatoire	20	
statut	État du rdv(confirmé, annulé)	A	obligatoire	20	
contrat_id	Code du contrat	AN	obligatoire	5	identifiant
montant	Prix fixé par le photographe et le client	AN	obligatoire	10	
date_creation	Date de signature du contrat	DATE	obligatoire	12	
statut_paiement	payé/avancé	A	obligatoire	20	

II.6.2 - Modèle Conceptuel de Données (MCD)

Le Modèle Conceptuel de Données (MCD) est un diagramme qui représente les entités du système, leurs attributs et les relations entre elles. Il permet de comprendre la structure des données et les interactions entre les différentes entités. Voici un extrait de notre MCD :

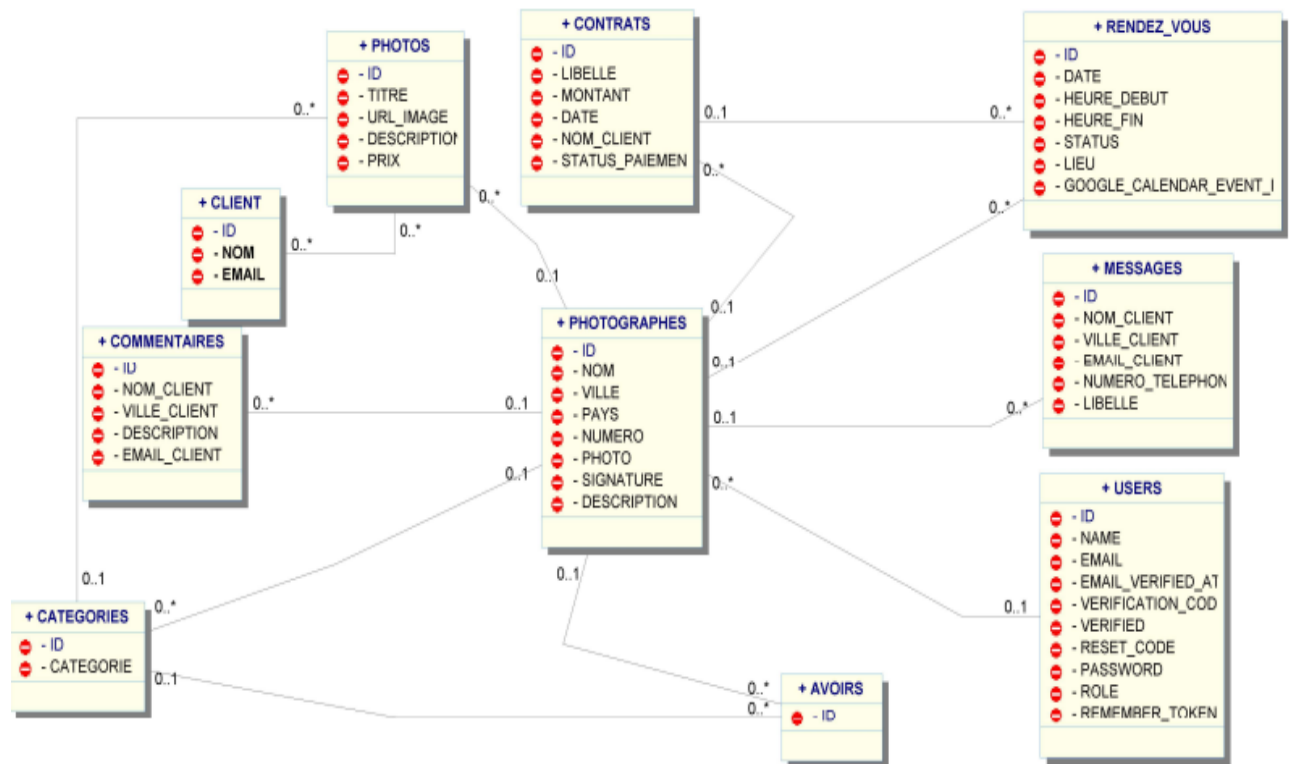


II.6.2 - Modèle Logique de Données (MLD)



II.7 - Diagramme de classe

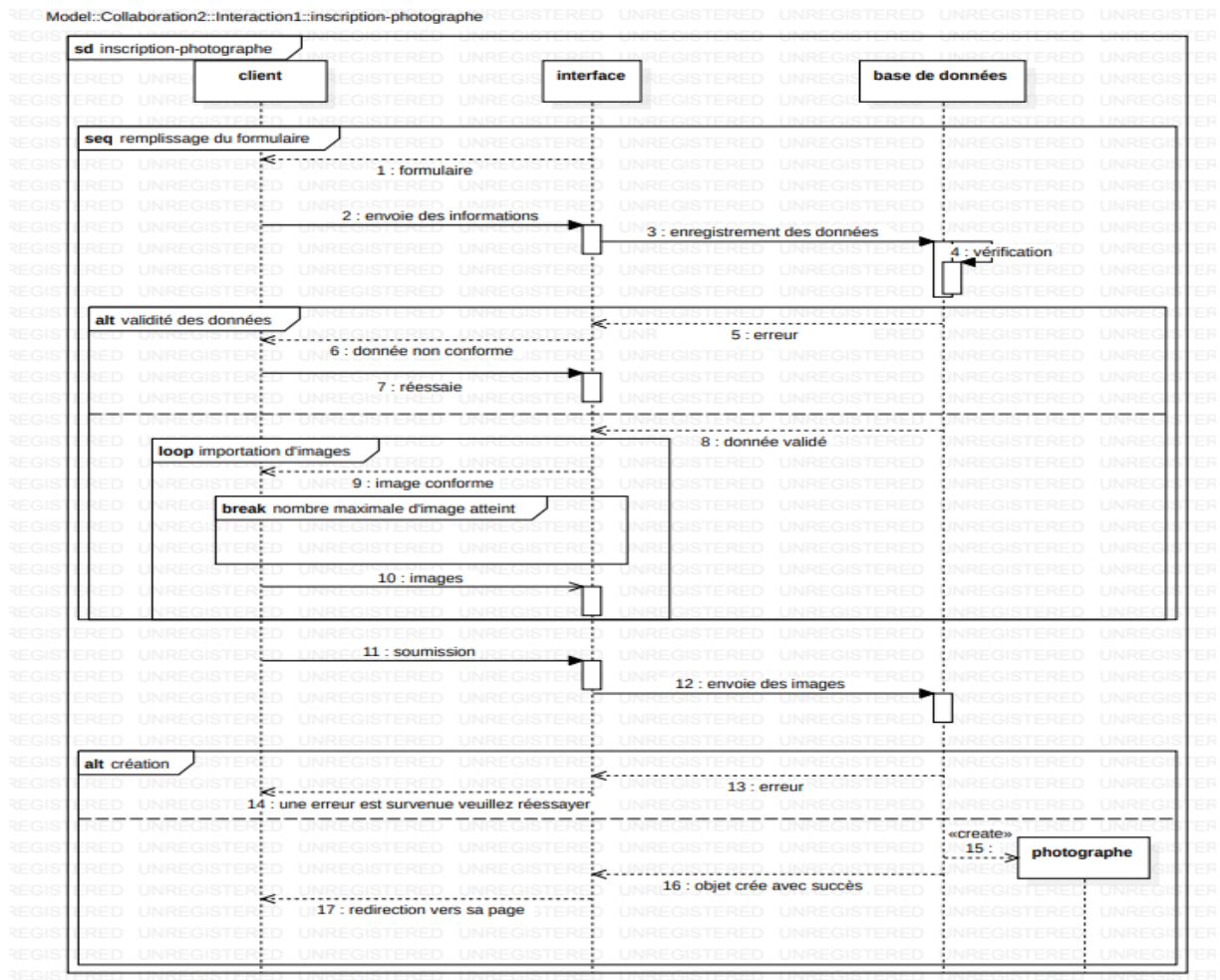
Le diagramme de classe est un outil essentiel pour la modélisation des structures de données et des relations entre les différentes entités du système. Il représente les classes du système, leurs attributs, leurs méthodes et les relations entre elles. Voici un tableau récapitulatif du diagramme de classe résultant:



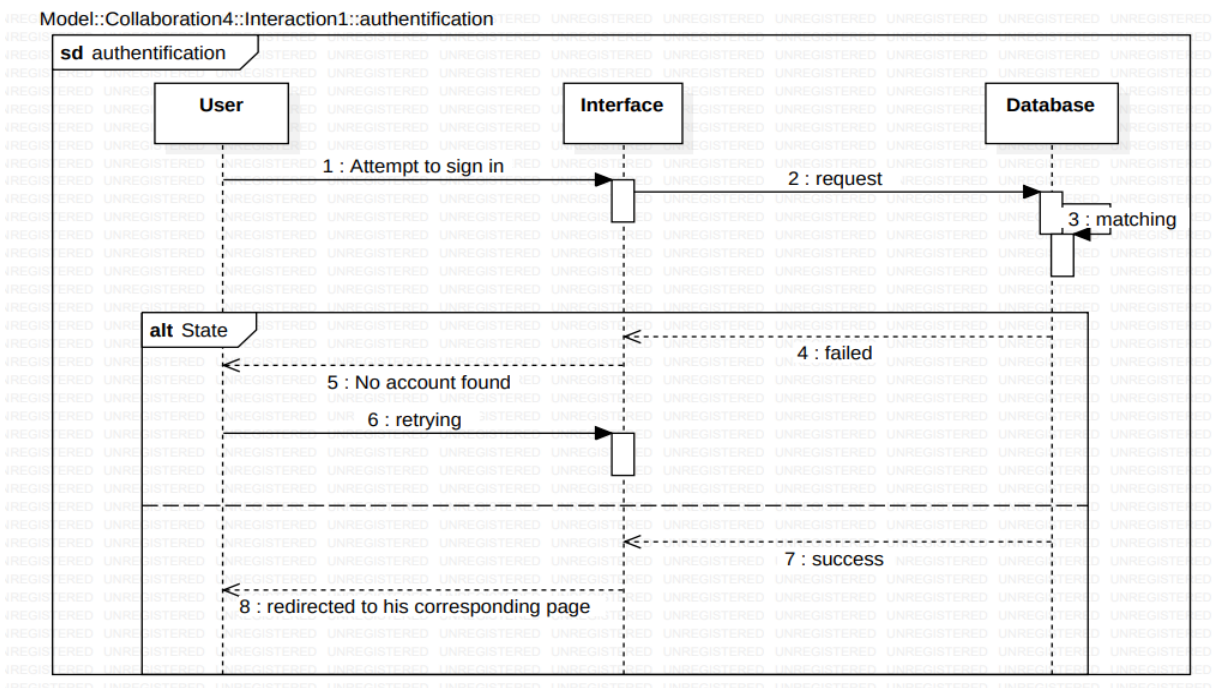
II.8 - Diagrammes de séquence

Les diagrammes de séquence sont utilisés pour modéliser les interactions entre les objets dans le système en fonction du temps. Ils montrent comment les objets communiquent entre eux en envoyant des messages. Nous avons créé des diagrammes de séquence pour quelques cas d'utilisation principal.

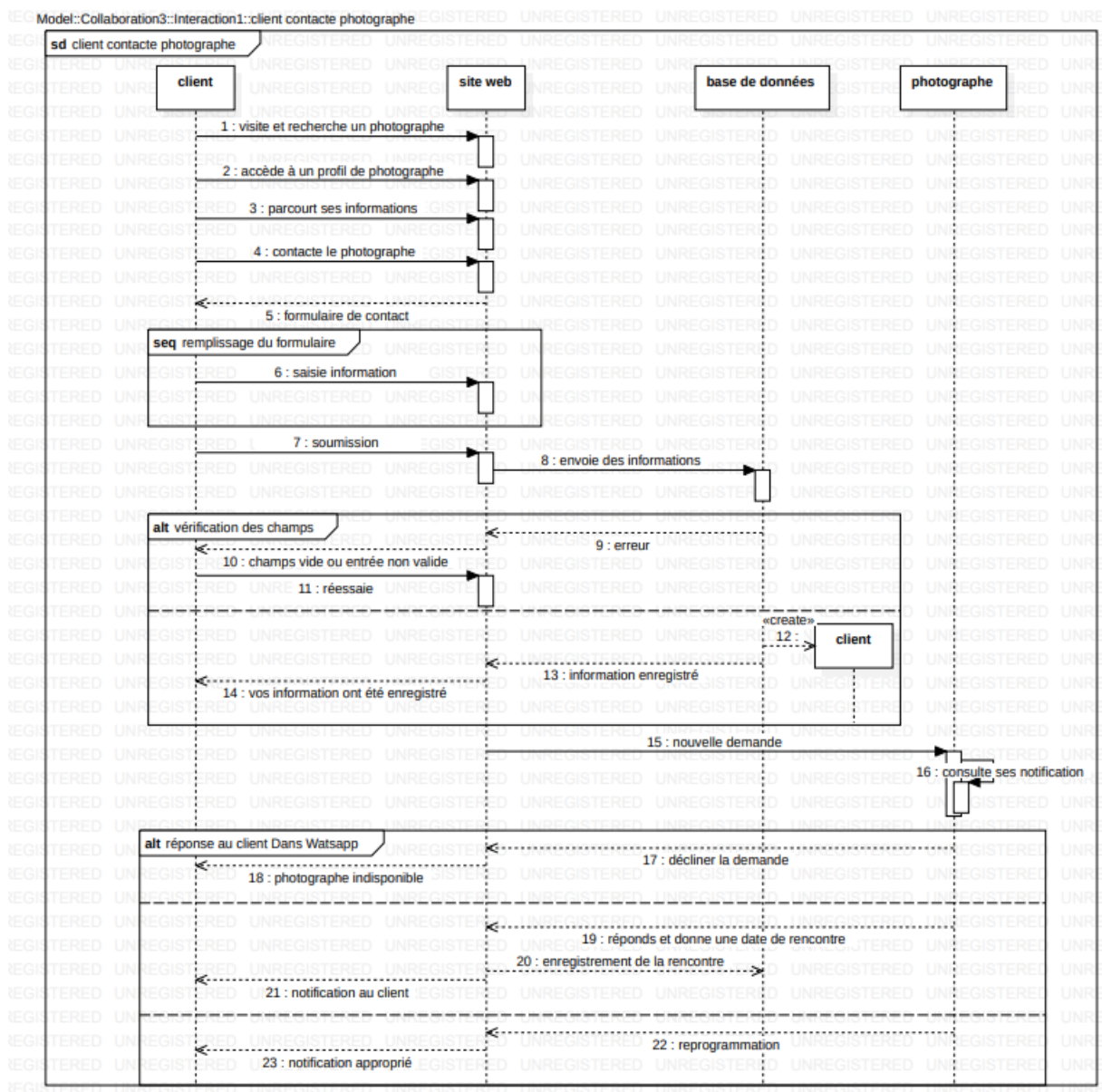
a-Diagramme séquence pour s'inscrire



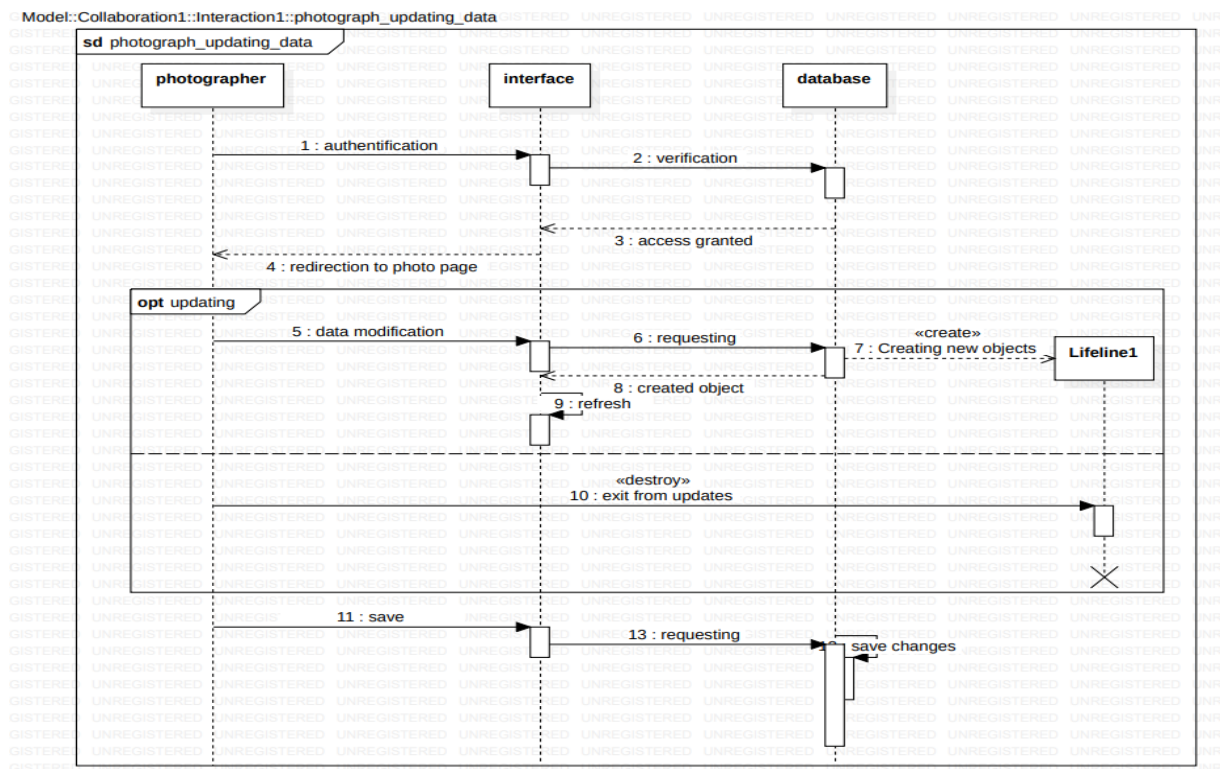
b-Diagramme pour s'authentifier



c-Diagramme pour contacter un photographe



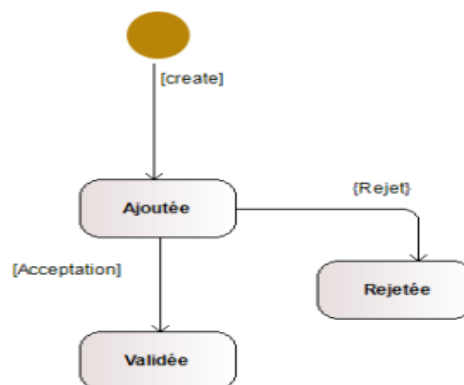
d-Diagramme pour modification de la donnée par le photographe



II.9 - Diagrammes d'état transition

Les diagrammes d'état transition sont utilisés pour représenter les différents états d'un objet et les transitions entre ces états en fonction des événements ou des conditions. Ils sont particulièrement utiles pour modéliser le comportement dynamique des objets.

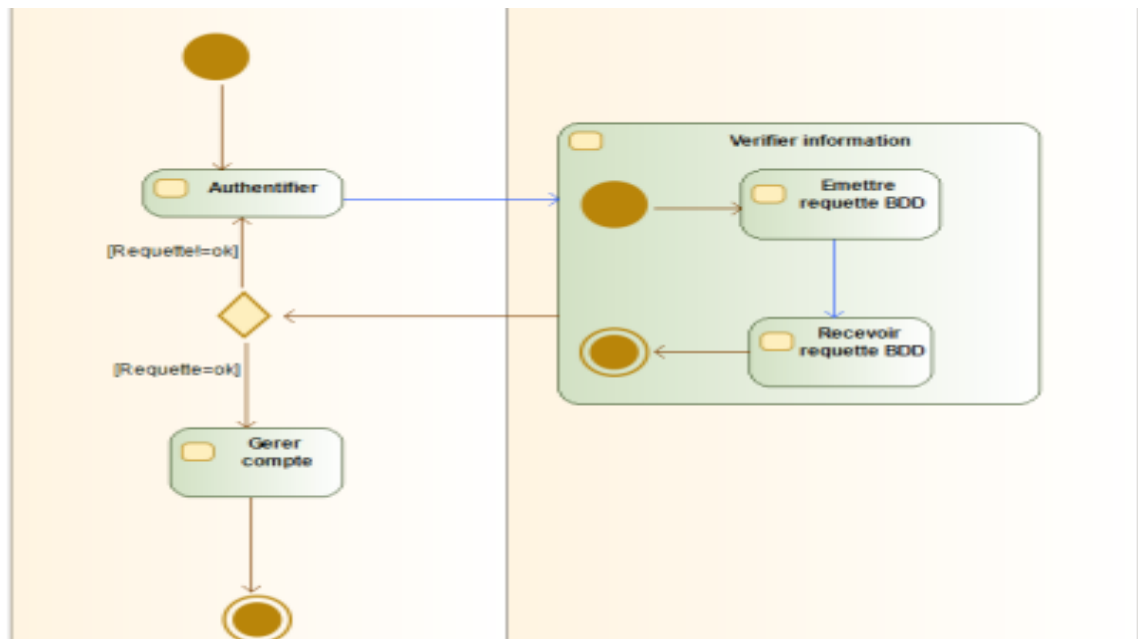
II.9.1 Diagramme d'état transition pour la photo



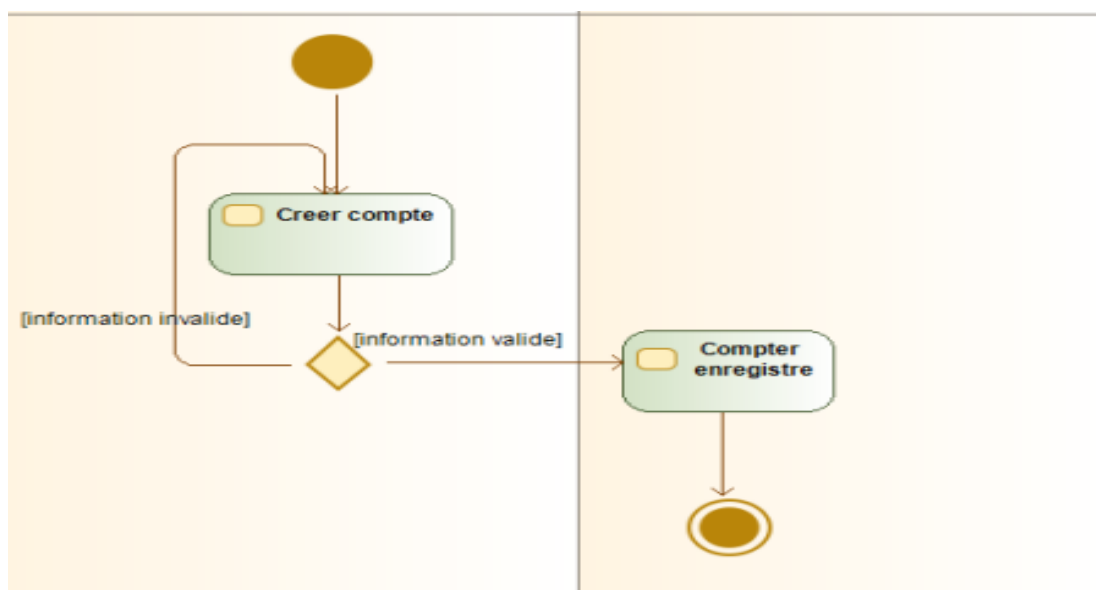
II.10 - Diagramme d'activités

Les diagrammes d'activités sont utilisés pour représenter les flux de contrôle et de données entre les différentes activités d'un système. Ils sont particulièrement utiles pour modéliser les processus d'affaires et les workflows.

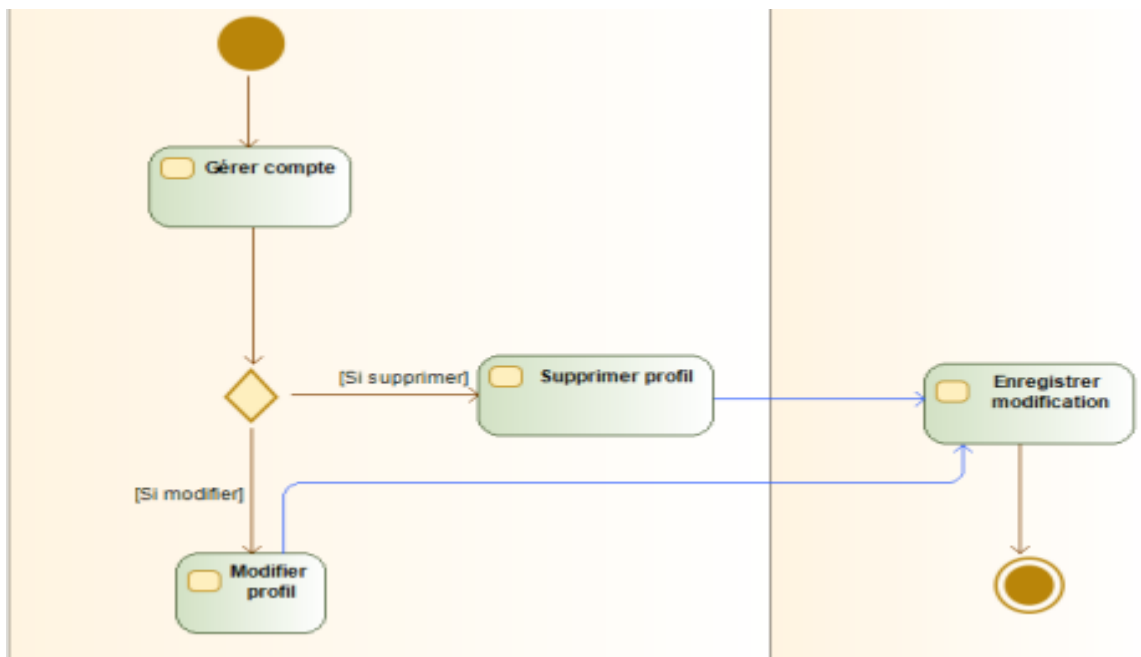
II.10.1 - Diagramme d'activités de S'authentifier



II.10.3 Diagramme d'activités de Créer un compte

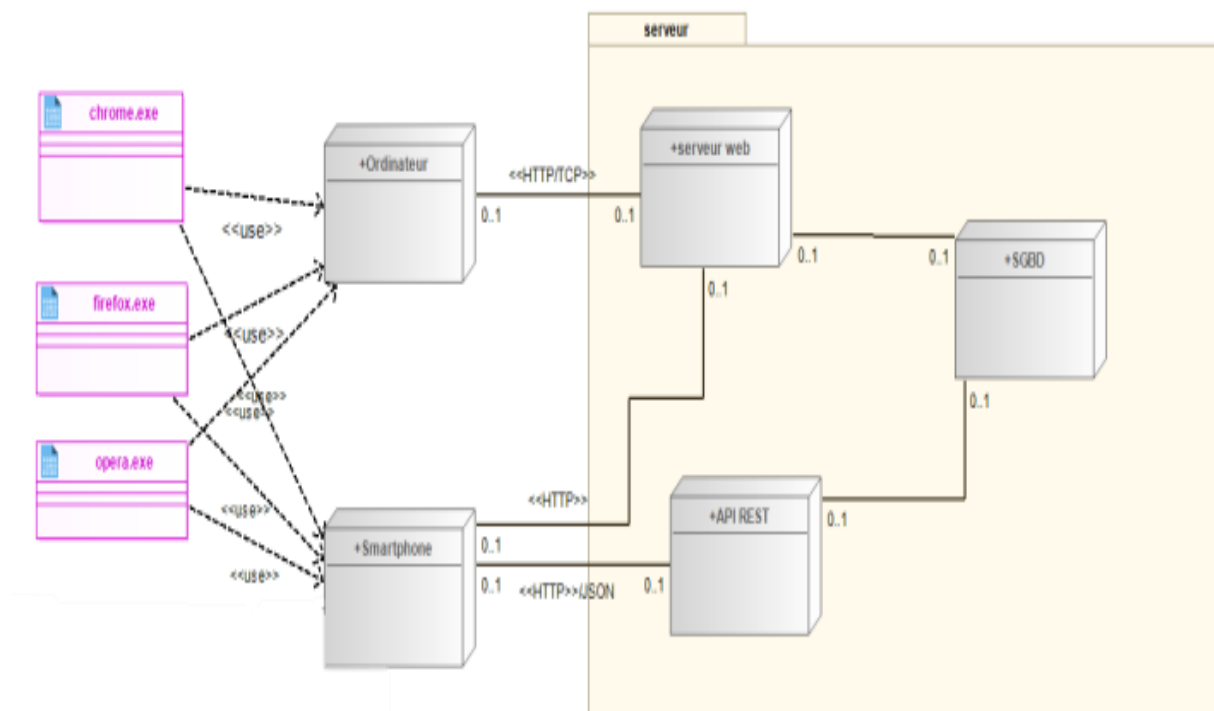


II.10.4 - Diagramme d'activités de Gérer un compte



II.11 - Diagramme de déploiement

Le diagramme de déploiement illustre l'architecture physique du système, y compris les nœuds physiques et les artefacts déployés sur ces nœuds. Dans le contexte de notre système, il montre comment les différents composants du système sont déployés sur l'infrastructure matérielle.



II.12 - Conclusion

En conclusion, l'analyse des algorithmes et des structures de données a été une étape cruciale dans le développement de notre système. Elle nous a permis de comprendre en profondeur les besoins du système, de définir les fonctionnalités clés et de concevoir l'architecture du système. L'utilisation de divers diagrammes UML et la mise en œuvre du modèle de conception MVC ont été essentielles pour structurer notre application de manière efficace et maintenable. Alors que nous passons à la phase de mise en œuvre, nous sommes confiants que l'analyse approfondie que nous avons effectuée nous aidera à développer un système robuste et efficace qui répondra aux besoins de nos utilisateurs.

Chapitre III : IMPLEMENTATION TECHNIQUE DE PROGRAMMATION ET RESULTATS

III.1 - Introduction


Dans ce chapitre, nous allons présenter les différentes étapes de l'implémentation de notre plateforme de réservation de logements en ligne. Nous allons décrire l'environnement technique que nous avons utilisé, les choix des outils et des technologies que nous avons faits, et les résultats que nous avons obtenus. Nous allons également expliquer les techniques de programmation que nous avons employées, ainsi que les difficultés que nous avons rencontrées et les solutions que nous avons apportées.

III.2 - Environnement technique (choix des outils)

Pour développer notre plateforme, nous avons utilisé un environnement technique composé de plusieurs outils et technologies. Nous avons choisi ces outils et technologies en fonction de leur pertinence, de leur performance, de leur popularité et de leur compatibilité. Voici une présentation des outils et technologies que nous avons utilisés :


III.2.1 - Présentation des technologies et Framework Utilisés

III.2.1.1 - HTML



HTML est un langage de balisage qui permet de structurer le contenu d'une page web. Il définit les éléments qui composent la page, tels que les titres, les paragraphes, les images, les liens, etc. Nous avons utilisé HTML pour créer la structure de base de notre interface utilisateur, en respectant les normes du W3C (World Wide Web Consortium)

III.2.1.2 - CSS



CSS est un langage qui permet de définir le style d'une page web, tels que les couleurs, les polices, les marges, etc. Il permet de séparer la présentation du contenu, ce qui facilite la maintenance et l'adaptabilité du site. Nous avons utilisé CSS pour donner un aspect visuel attrayant à notre interface utilisateur, en suivant les principes du responsive design (adaptation aux différentes tailles d'écran)

III.2.1.3 - Laravel



Laravel est un Framework PHP qui offre une structure robuste et flexible pour la création d'applications web modernes. Il offre une variété de fonctionnalités qui facilitent le développement d'applications web, y compris une architecture MVC (Model-View-Controller), un système de routage facile à utiliser, une intégration facile avec les bases de données, et une sécurité robuste. Nous avons choisi Laravel comme Framework pour développer notre plateforme car il offre plusieurs avantages, tels que :

- **Facilité de développement** : Laravel offre une syntaxe élégante et expressive qui facilite le développement des applications web.
- Il offre une documentation complète et une grande communauté de support
- **Intégration avec d'autres services** : Laravel offre une intégration avec d'autres services tels que les services de paiements en ligne, les API externes. Cela permet d'étendre les fonctionnalités de la plateforme en intégrant des services tiers pertinents pour les photographes.

III.2.1.4 - Angular



Angular est un Framework open-source développé par Google qui permet de créer des applications web dynamique et interactives. Il utilise le langage TypeScript et offre une architecture basée sur des composants, ce qui facilite la création d'interfaces utilisateur réactives

III.2.1.5 - Spring Boot



Spring Boot est un Framework java qui simplifie le développement d'application web en fournissant une configuration défaut et en intégrant de nombreux composants de la suite Spring. Il facilite la création de micro services et d'applications RESTful en offrant une mise en route rapide et une gestion simplifiée des dépendances

III.2.1.6 - UML



UML est un langage de modélisation qui permet de représenter graphiquement les différents aspects d'un système informatique. Il utilise des diagrammes pour décrire la structure, le comportement et les interactions du système. Nous avons utilisé UML pour concevoir et analyser notre système, en utilisant des diagrammes tels que les diagrammes de cas d'utilisation, de classes et de séquences

III.2.1.7 - MERISE



MERISE est une méthode d'analyse et de conception des systèmes d'information basée sur la séparation des données et des traitements. Elle utilise des modèles pour représenter les données, les traitements et les interactions du système. Nous avons utilisé MERISE pour concevoir et analyser notre base de données, en utilisant des modèles tels que le Modèle Conceptuel de Données (MCD) et le Modèle Logique de Données (MLD).

III.2.2 - Environnement de travail

III.2.2.2 - VS Code



VS Code est un éditeur de code source qui offre une variété de fonctionnalités pour le développement web, telles que la coloration syntaxique, l'auto-complétion, le débogage, etc. Nous avons utilisé VS Code pour écrire le code Angular, Laravel, HTML, CSS de notre plateforme, en profitant des extensions disponibles pour Laravel etc.

III.2.2.3 - Chrome



Chrome est un navigateur web qui offre une performance élevée et une compatibilité avec les standards du web. Nous avons utilisé Chrome pour tester notre plateforme, en utilisant les outils de développement intégrés pour inspecter le code, vérifier les erreurs, mesurer la performance, etc

III.2.2.4 - Git/Git Hub



Git est un système de contrôle de version qui permet de gérer les modifications du code source d'un projet. Git Hub est une plateforme en ligne qui permet d'héberger des projets basés sur Git et d'offrir des fonctionnalités telles que la collaboration, la revue de code, la gestion des problèmes, etc. Nous avons utilisé Git/Git Hub pour gérer le code source de notre plateforme, en créant des branches, des commit. Le lien de notre projet est disponible à ce lien : https://github.com/regent-ndogmo/UV_projet.git

III.2.2.5 - Android Studio



Android Studio est un environnement de développement intégré (IDE) qui permet de créer des applications mobiles pour Android. Il offre une variété de fonctionnalités qui facilitent le développement d'applications mobiles, y compris un éditeur de code intelligent, un émulateur Android, un débogueur, etc. Nous avons utilisé Android Studio pour développer notre application mobile, en utilisant le langage Java et le SDK (Software Development Kit) Android.

III.2.2.6- PostgreSQL



PostgreSQL est un système de gestion de base de données open-source très puissant et fiable. Il supporte les fonctionnalités avancées tels que, les vues, les procédures stockées, etc. Nous avons utilisé PostgreSQL la gestion de notre base de données.

III.2.2.7 - Postman



Postman est un outil qui permet de tester et d'interagir avec des API (Application Programming Interface) web. Il offre une interface graphique qui permet d'envoyer des requêtes HTTP (Hypertext Transfer Protocol) et d'afficher les réponses du serveur. Nous avons utilisé Postman pour tester l'API RESTful que nous avons créée pour notre plateforme, en vérifiant le format JSON (JavaScript Object Notation) des données échangées.

III.2.2.8- Serveurs de développement

Nous avons utilisé deux serveurs de développement pour héberger notre plateforme :

- ✚ Un serveur local sur notre machine, en utilisant Laragon (un ensemble de logiciels comprenant Apache, MySQL, PHP). Nous avons utilisé ces serveurs pour tester le fonctionnement de notre plateforme en local, et pour faciliter la collaboration entre les membres du groupe.

III.2.2.9 - Systèmes d'exploitation (Windows 10 et Ubuntu)

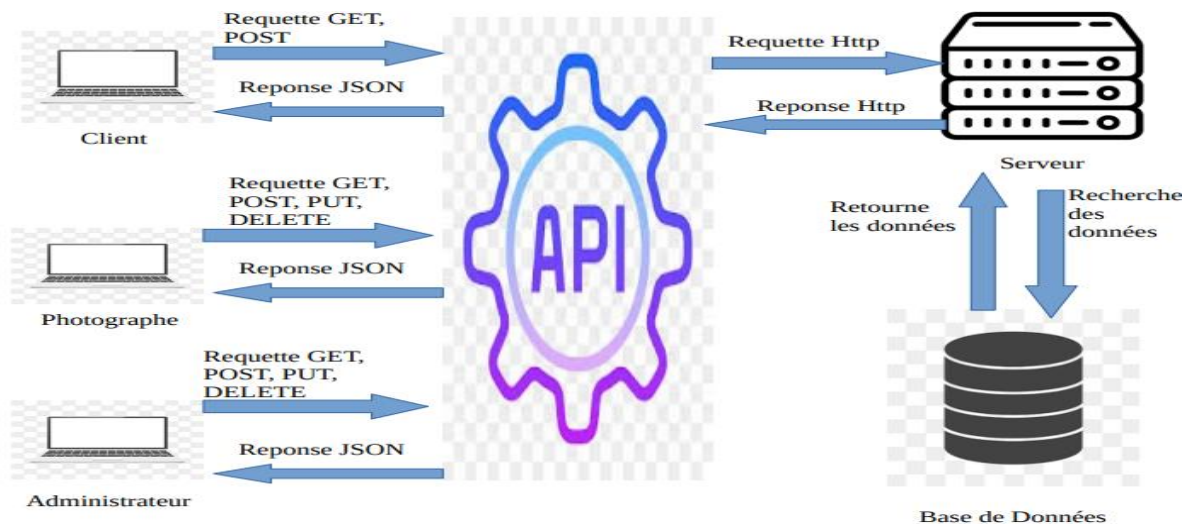
Nous avons utilisé différents systèmes d'exploitation pour développer notre plateforme:

- ✚ Windows 10 qui est un système d'exploitation populaire et largement utilisé pour le développement web.
- ✚ Ubuntu, qui est un système d'exploitation basé sur Linux et qui offre une stabilité et une sécurité élevées.

III.4 - Architecture de notre plateforme

Pour la conception et la réalisation de notre plateforme dédiée aux photographes, nous allons adopter une architecture client-serveur. Cette architecture fonctionnera de la manière suivante :

- **Demande du client** : les clients sont des programmes exécutés sur les ordinateurs des utilisateurs finaux (Photographes, Clients, Administrateurs). Un client envoie une demande à un serveur pour obtenir un service ou des données. La demande peut être envoyée via un réseau local ou Internet.
- **Traitement de la demande par le serveur** : Le serveur reçoit la demande du client et la traite. Cela peut impliquer l'accès à une base de données, l'exécution de calculs complexes ou la récupération de fichiers.
- **Réponse du serveur** : Le serveur envoie une réponse au client. La réponse peut inclure les données demandées, le résultat d'un calcul ou un message d'erreur.
- **Réception de la réponse par le client** : Le client reçoit la réponse du serveur et l'affiche à l'utilisateur. Cette architecture peut être représentée de la manière suivante :



III.5 - Présentation de quelques résultats clés et Algorithmes d'obtention des résultats

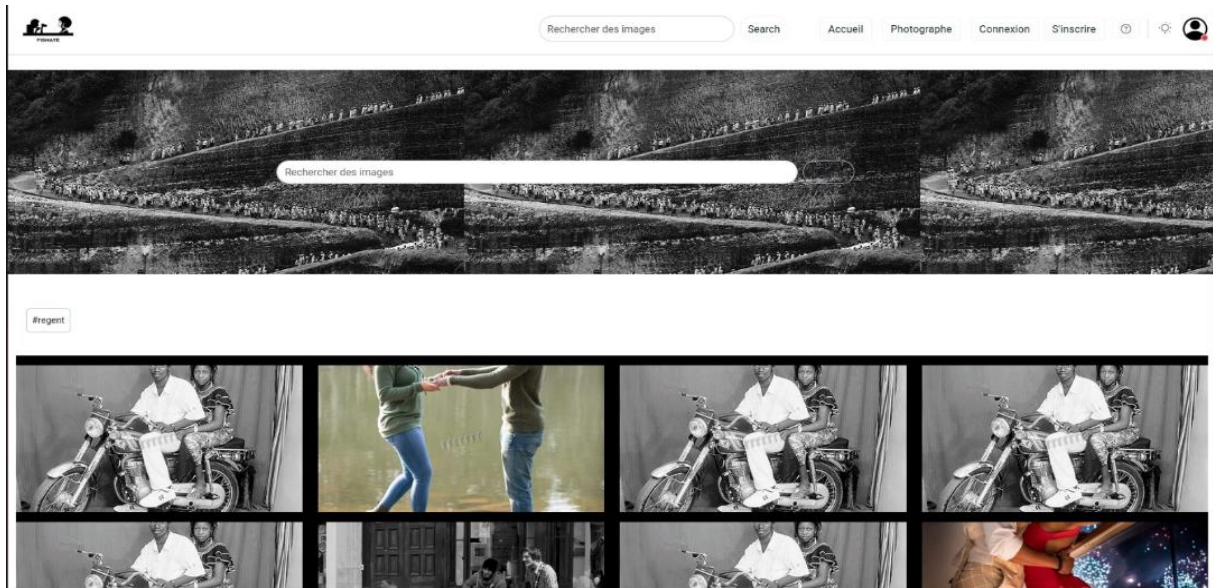
Dans cette section, nous allons présenter quelques résultats clés que nous avons obtenus grâce à l'implémentation de notre plateforme. Nous allons également expliquer les algorithmes que nous avons utilisés pour obtenir ces résultats.

III.5.1 - Front end

Le Frontend est la partie de la plateforme qui interagit avec l'utilisateur. Il comprend l'interface utilisateur, qui est composée de pages web dynamiques créées avec HTML, CSS et le Framework JavaScript angularJs pour produire l'interface utilisateur (UI) et les vues adéquates propres aux utilisateurs. Le Frontend communique avec le backend via des requêtes HTTP. Voici quelques captures d'écran de notre interface utilisateur :

III.5.1.1 - Interface d'accueil

Cette page présente le nom et le logo de notre plateforme, ainsi qu'une barre de recherche qui permet à l'utilisateur de rechercher un photographe en fonction de la localisation, la catégorie de photo, le type de photos. Elle affiche également quelques photos publiées par les photographes ainsi que le menu d'accueil, photographe qui regroupe tous les photographes de notre plateforme, connexion et déconnexion qui permettent respectivement aux utilisateurs de se connecter et de se déconnecter.



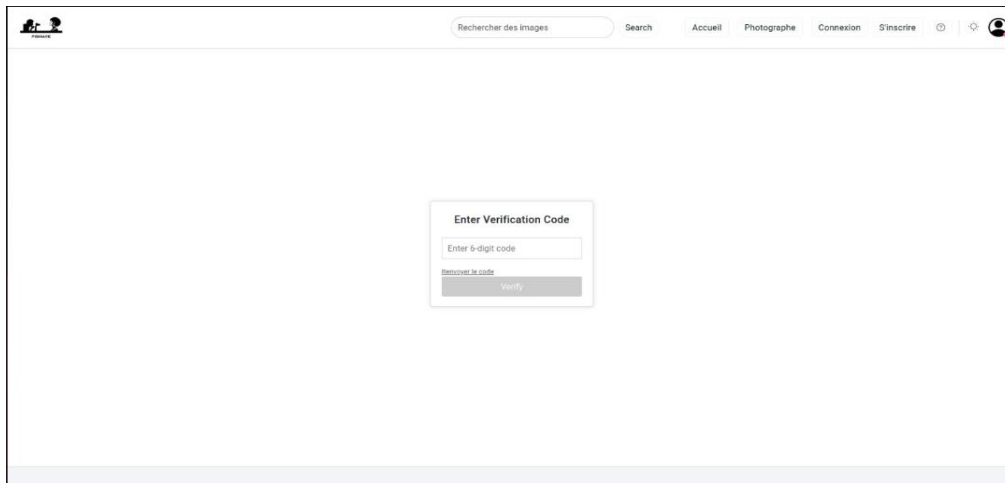
III.5.1.2 - Interface pour inscription du photographe

Cette interface concerne tout utilisateur qui voudrait s'inscrire comme photographe de notre plateforme. Elle contient les zones de remplissage des informations personnelles de l'utilisateur.

The screenshot shows the registration page for a photographer on the FISHAYE platform. The page has a light gray background. At the top, there is a navigation bar with the platform's logo on the left and a search bar labeled 'Rechercher des images' in the center. To the right of the search bar are links for 'Search', 'Accueil', 'Photographe', 'Connexion', 'S'inscrire', and a user profile icon. Below the navigation bar, there is a large white box with the title 'S'enregistrer'. Inside this box, there are four input fields for 'Votre nom', 'Votre Email', 'Mot de Passe', and 'Confirmer le mot de passe'. Below these fields is a checkbox labeled 'J'accepte les termes et conditions'. At the bottom of the box is a blue 'Register' button. To the right of the input fields is a colorful illustration of a person holding a camera, with various geometric shapes and colors around them.

III.5.1.3 - Interface pour envoyer le code de vérification

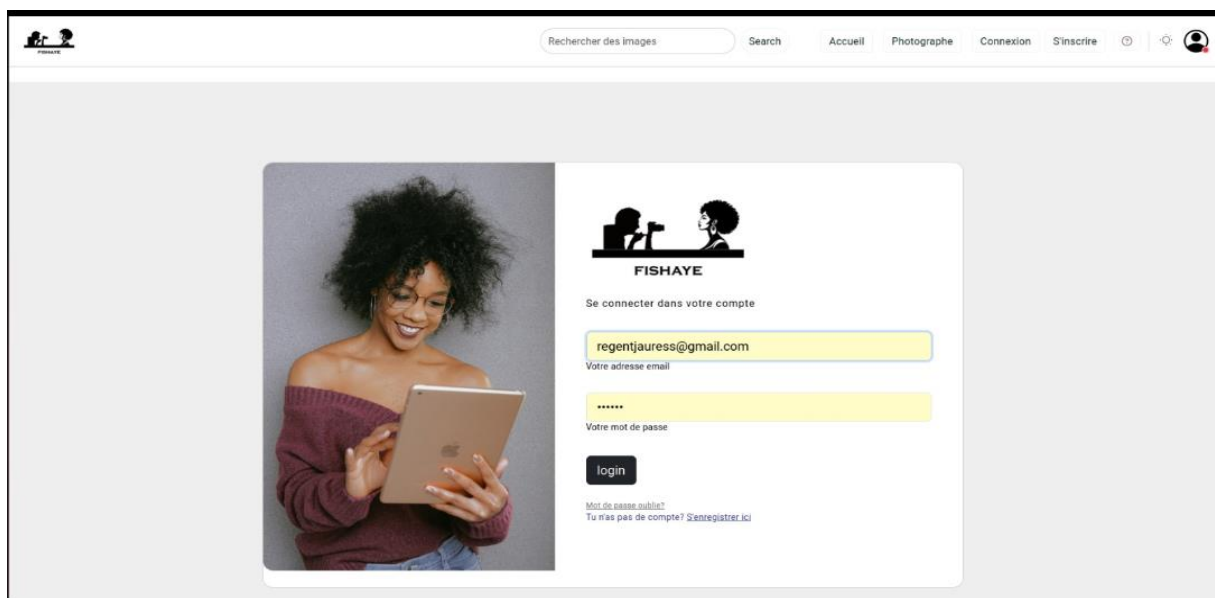
Le code de vérification permet aux utilisateurs de faire une double authentification en vérifiant l'email renseigné. Ils peuvent ainsi garantir leur sécurité. Ce code est envoyé à l'adresse mail renseignée par l'utilisateur, il comporte 6 chiffres qui seront invalides après une durée d'une minute. L'utilisateur doit cliquer sur renvoyer le code pour le recevoir à nouveau.



The screenshot shows a web browser window with a navigation bar at the top containing a logo, a search bar with the placeholder 'Rechercher des images', and links for 'Search', 'Accueil', 'Photographe', 'Connexion', and 'S'inscrire'. In the center of the page is a modal box titled 'Enter Verification Code'. Inside this box, there is a text input field labeled 'Enter 6-digit code', a link that says 'Renvoyer le code', and a 'Verify' button.

III.5.1.4 - Interface pour connexion

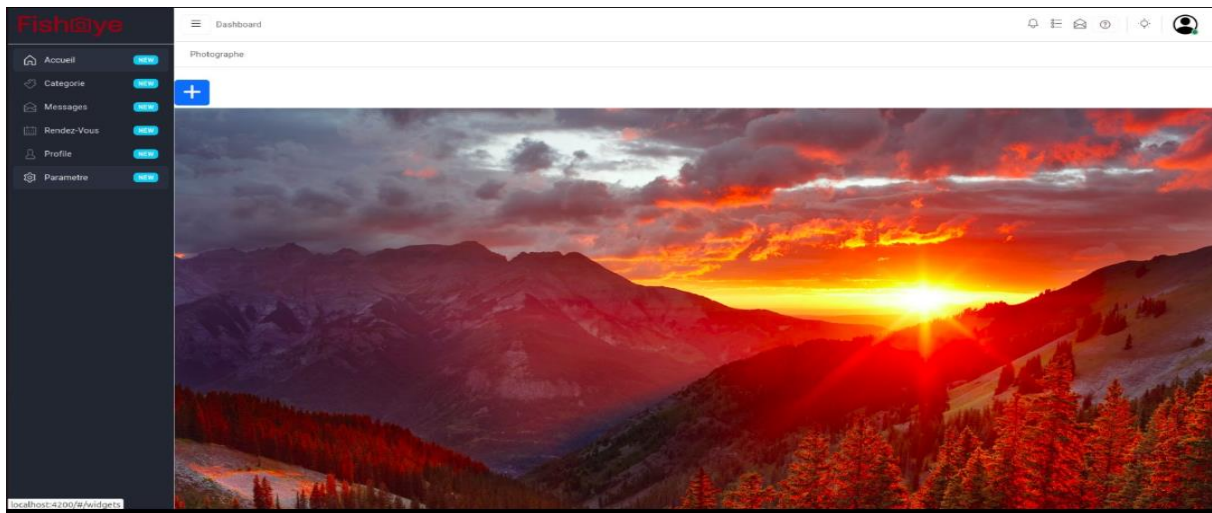
Cette interface permet à un photographe de se connecter à la plateforme pour accéder aux services offerts.



The screenshot displays the FISHAYE login page. The navigation bar at the top is identical to the previous one. The main content area features a large image of a smiling woman with curly hair holding a tablet. To the right of the image is a white login form. The form has the FISHAYE logo at the top, followed by the text 'Se connecter dans votre compte'. It contains two input fields: one for the email address, which is pre-filled with 'regentjauress@gmail.com', and another for the password, shown with asterisks. Below these fields is a 'login' button. At the bottom of the form, there is a link that reads 'Mot de passe oublié? Tu n'as pas de compte? S'inscrire ici'.

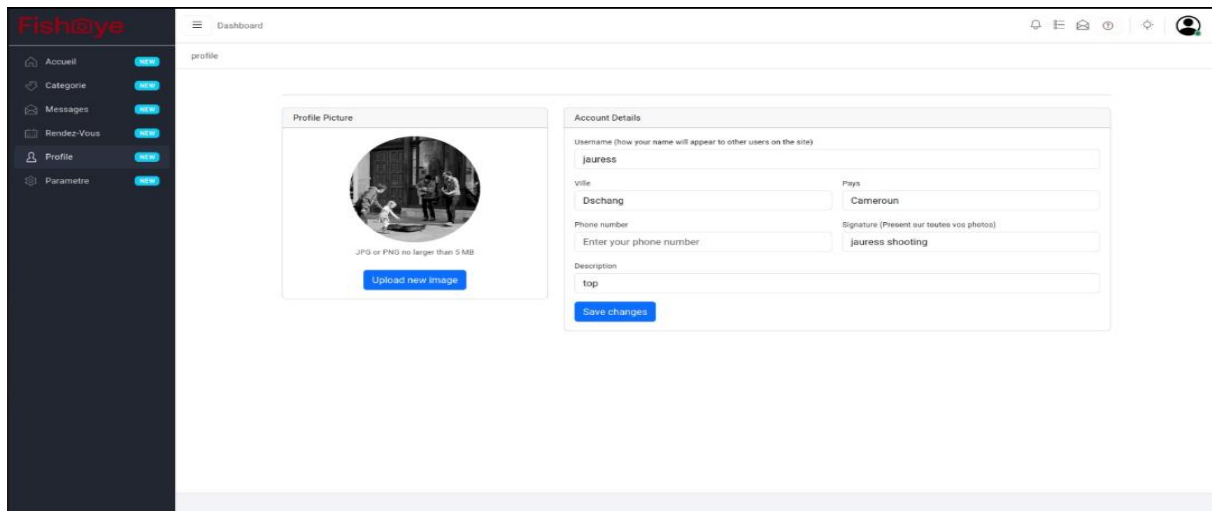
III.5.1.5 - Interface pour accueil photographe

Cette interface s'affiche après la connexion du photographe.



III.5.1.6 - Interface de Dashboard du photographe

Cette page permet au photographe de gérer son profil. Elle présente, sa photo de profil, les détails de son compte comportent : son nom, sa ville son numéro de téléphone, la description et la signature de ses photos



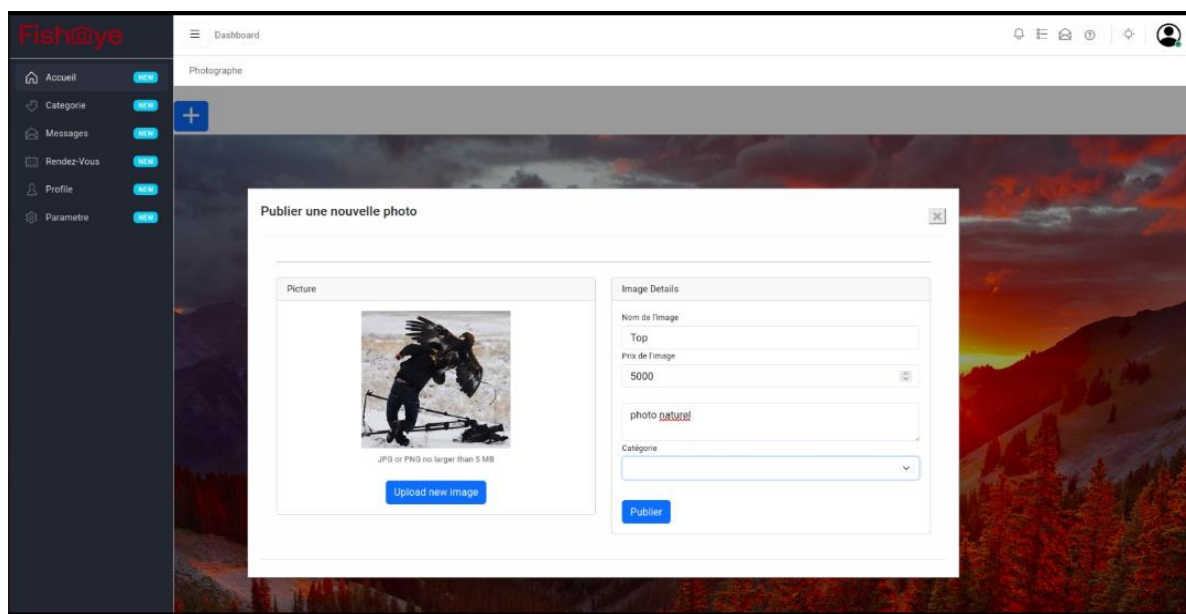
III.5.1.7 - Interface pour créer une catégorie

Cette interface permet au photographe de créer une catégorie dans laquelle il va publier ses photos concernant le type de catégorie créée



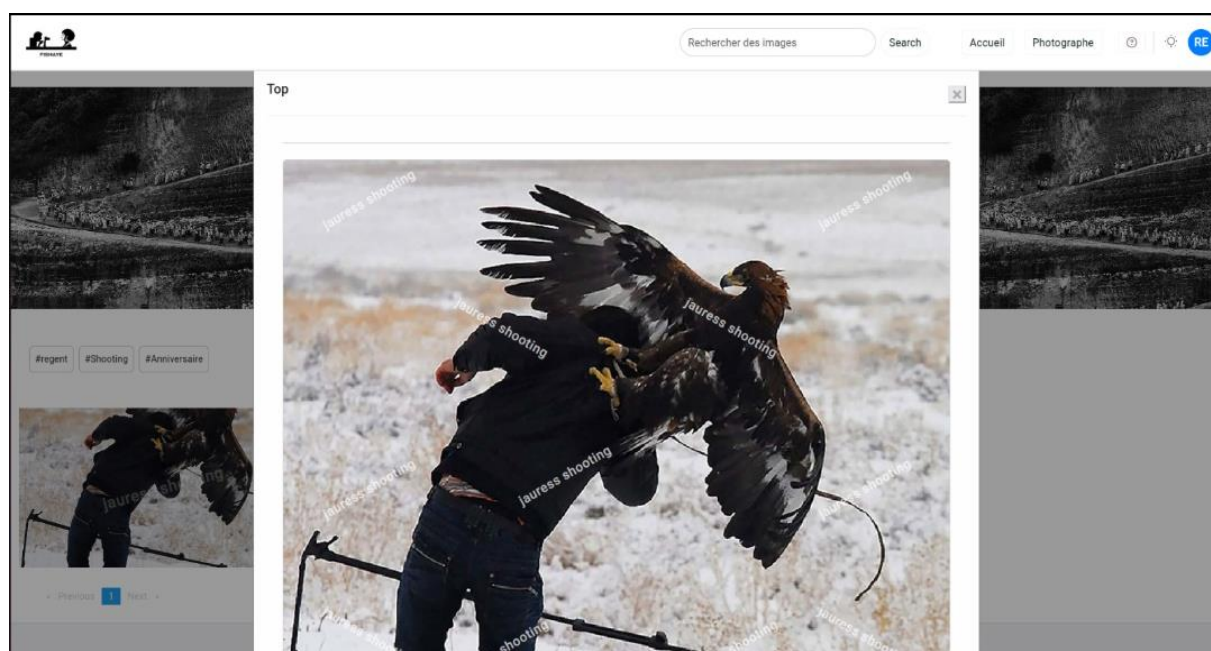
III.5.1.8 - Interface pour publier une photo dans une catégorie

Cette interface contient un formulaire contenant l'information de la photo à savoir : le nom de l'image, prix de l'image (facultatif), la description et la catégorie de l'image.



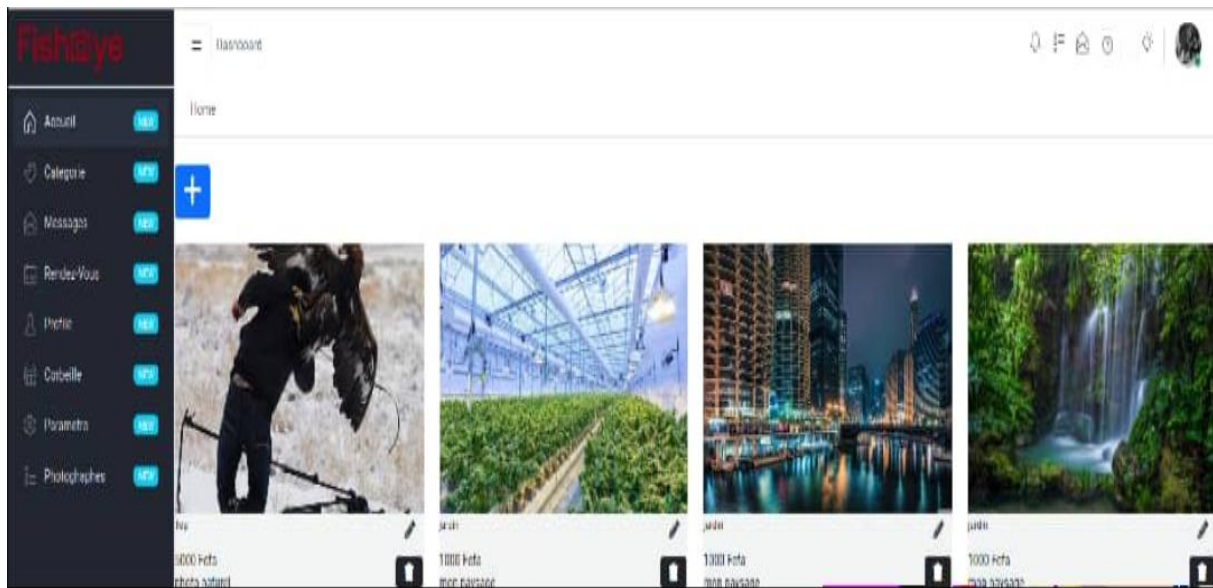
III.5.1.9 - Interface pour signature de l'image

Après avoir publié l'image dans une catégorie, celle-ci prend automatiquement le nom de la signature que vous avez défini sur votre profil.



III.5.1.10- Interface d'administration

Cette page permet à l'administrateur du système de gérer l'ensemble du système.



III.5.2 - Backend

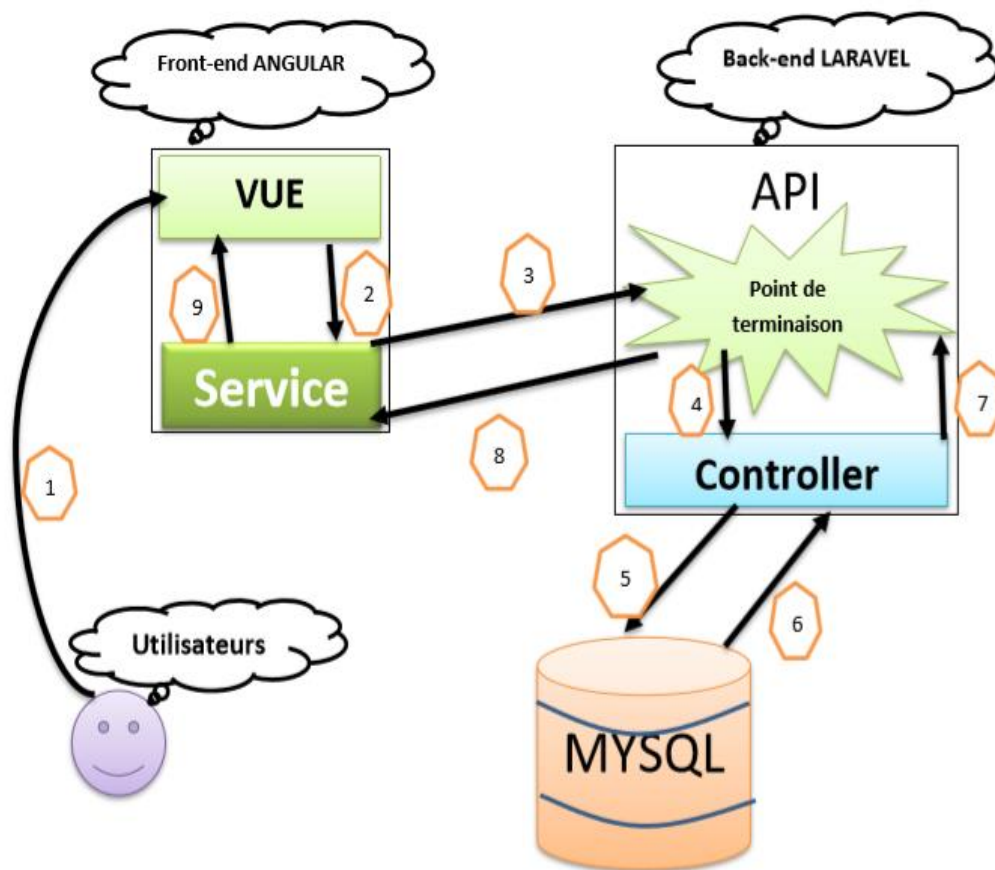
Le backend est la partie de la plateforme qui traite les données et la logique du système. Il comprend la base de données, qui est composée de tables relationnelles stockées dans PostgreSQL, et l'API RESTful, qui est composée de routes PHP qui implémentent les fonctionnalités du système. Le backend communique avec le Frontend via des requêtes HTTP et renvoie des données au format JSON.

III.5.3 - Communication entre Frontend et Backend

Le Frontend permettra aux utilisateurs (clients, photographes et administrateurs) de point d'entrée à la plateforme. Ce dernier à travers des services spécifiques pourra accéder au backend Laravel via une API RESTful qui servira de commutateur pour l'échange Des données au format JSON. Quant à celui-ci il est directement lié à un support de stockage (BD) et permet des interactions via des requêtes SQL à l'aide du SGBD PostgreSQL : Ainsi, le mécanisme de communication est le suivant :

1) L'utilisateur accède à la plateforme via son équipement (mobile ou pc) et le Frontend affiche la vue correspondante et lance une demande de données (par exemple pour la consultation, insertion, authentification etc.)

- 2) La demande est acheminée via un service angular vers le point de terminaison considéré
- 3) Ce service fait alors appel l'API RESTful pour l'usage du Backend
- 4) Cette API est redirigé vers le contrôleur correspondant à la demande du service
- 5) Ce contrôleur fait à son tour une requête SQL pour accéder au donnée via MySQL
- 6) Ce dernier répond selon le type de requête (CRUD généralement)
- 7) Celui-ci, cette fois se charge de formater la réponse en JSON qui est renvoyé au service
- 8) Ce service grâce aux observateurs récupère la réponse et l'envoi finalement à la vue
- 9) La réponse est renvoyée à l'UI



Chapitre IV : APPORT, LEÇON, DIFFICULTÉS ET SOLUTIONS

IV.1 - Introduction

Dans ce chapitre, nous allons faire le bilan de notre projet. Nous allons présenter les apports et les leçons que nous avons tirés de cette expérience, ainsi que les problèmes que nous avons rencontrés et les solutions que nous avons apportées.

IV.2 - Apports et leçons tirées du projet

IV.2.1 - Apport pour nous

Ce projet a été pour nous une opportunité d'apprendre et de mettre en pratique des concepts et des technologies liés au développement web. Nous avons pu acquérir des compétences techniques, telles que :

- La maîtrise du langage du Framework Laravel, qui nous a permis de créer une API RESTful robuste et flexible pour notre plateforme.
- La maîtrise du langage HTML, CSS, et le Framework Angular, qui nous ont permis de créer une interface utilisateur intuitive et responsive pour notre plateforme.
- La maîtrise du langage SQL et du système de gestion de bases de données PostgreSQL, qui nous ont permis de stocker et de manipuler les données de notre plateforme.
- La maîtrise du langage Java et de l'environnement de développement Android Studio, qui nous ont permis de créer une application mobile pour notre plateforme.
- La maîtrise des outils et logiciels tels que VS Code, Git/GitHub, Postman, etc., qui nous ont facilité le développement, le test de notre plateforme.

D'autre part ce projet nous a été bénéfique dans l'optique où il nous a permis de:

- Analyser les besoins des utilisateurs et à concevoir un cahier des charges adapté.
- Modéliser un système informatique avec des diagrammes UML et MERISE.
- Travailler en équipe et à collaborer efficacement avec les autres membres du groupe.

IV.2.2 - Apport pour la communauté

- ✚ Notre plateforme dédiée aux photographes peut apporter de nombreux avantages à la communauté des photographes, notamment :
- ✚ Partage et visibilité : Notre plateforme offre aux photographes un espace pour partager leurs travaux avec un public plus large, ce qui peut contribuer à accroître leur visibilité et à promouvoir leur talent.
- ✚ Réseautage et collaboration : Les photographes peuvent utiliser la plateforme pour se connecter avec d'autres professionnels du secteur, échanger des idées, collaborer sur des projets et établir des partenariats.
- ✚ Opportunités professionnelles : notre plateforme peut servir de vitrine pour les photographes en recherche de clients potentiels, d'opportunités de travail ou de contrats
- ✚ Promotion de l'art de la photographie : En offrant un espace dédié à la photographie, notre plateforme contribue à promouvoir l'appréciation de l'art de la photographie et à sensibiliser le public à la créativité et au talent des photographes.

En plus des avantages pour la communauté des photographes, Notre plateforme dédiée aux photographes apporte également des avantages significatifs à la population en général :

- ✚ Accès à l'art et à la créativité : Notre plateforme offre au public un accès facile à une grande variété d'œuvres photographiques, ce qui peut contribuer à élargir l'appréciation de l'art de la photographie et à enrichir la culture visuelle de la population.
- ✚ Sensibilisation aux enjeux sociaux et environnementaux : Les photographes utilisent souvent leur art pour sensibiliser le public à des questions importantes telles que les enjeux sociaux, environnementaux ou politiques. Notre plateforme dédiée aux photographes peut servir de vitrine pour ces sujets et encourager le dialogue et la sensibilisation.
- ✚ Promotion du tourisme et du patrimoine : Les photographies de paysages, de lieux historiques et de destinations touristiques peuvent contribuer à promouvoir le tourisme local et à mettre en valeur le patrimoine culturel et naturel d'une région.

IV.3 - Problèmes rencontrés et quelques solutions

Tout au long du projet, nous avons rencontré plusieurs problèmes, qui ont constitué des défis à relever. Voici quelques exemples de problèmes que nous avons rencontrés et les solutions que nous avons apportées :

- Un problème lié au choix des technologies et des outils : Nous avons dû choisir les technologies et les outils les plus adaptés à notre projet, en tenant compte de leur pertinence, de leur performance, de leur popularité et de leur compatibilité. Nous avons effectué une recherche préalable sur les différentes options disponibles, et nous avons comparé leurs avantages et leurs inconvénients. Nous avons également consulté des tutoriels, des documentations et des forums pour nous aider à prendre notre décision.
- Un problème lié à l'intégration des technologies et des outils : Nous avons dû faire en sorte que les technologies et les outils que nous avons choisis fonctionnent ensemble de manière cohérente et harmonieuse. Nous avons dû résoudre des problèmes de compatibilité, de configuration, de dépendance, etc. Nous avons utilisé des solutions telles que Composer, NPM, etc., pour gérer les dépendances et les paquets de notre projet.
- Un problème lié à la conception et à l'analyse du système : Nous avons dû concevoir et analyser notre système de manière rigoureuse et structurée, en respectant les normes et les bonnes pratiques. Nous avons dû définir les besoins fonctionnels et non fonctionnels du système, modéliser le système avec des diagrammes UML et MERISE, définir le dictionnaire de données et le Modèle Conceptuel de Données (MCD).
- Un problème dû à l'intégration de l'api de Google calendar pour la gestion des rendez-vous par un photographe : Nous avons dû intégrer l'api de Google calendar pour qu'un photographe puisse gérer ces différents événements de manière sécurisée et structurée.

CONCLUSION GENERALE

En conclusion, la création d'une plateforme dédiée aux photographes représente une avancée significative dans le domaine de la photographie. Cette plateforme offre un espace centralisé pour les photographes professionnels et amateurs afin de partager leurs travaux, d'échanger des conseils et des expériences, et de se connecter avec des clients potentiels. En outre, elle facilite l'accès à des ressources et des outils utiles pour le développement professionnel des photographes. La mise en place d'une telle plateforme nécessite une approche soigneusement étudiée, prenant en compte les besoins spécifiques de la communauté des photographes et les avancées technologiques actuelles. En fin de compte, cette plateforme peut contribuer de manière significative à l'évolution et à la promotion de l'art de la photographie dans un monde de plus en plus numérique. Notre plateforme, réalisée avec Laravel, HTML, CSS, Springboot, Ionic, PostgreSQL, Android Studio, entre autres outils, a suivi une approche itérative et incrémentale. Une analyse détaillée des besoins a précédé la modélisation du système via des diagrammes UML et MERISE, avant la mise en œuvre de l'architecture MVC respectant les principes du développement web. Malgré que nous n'ayons pas pu déployer notre application, Nous sommes fiers du travail accompli jusqu'ici. Notre espoir est que les utilisateurs apprécient notre plateforme dédiée aux photographes et y trouvent des photographes selon leurs désirs.

Lien github pour le code source : https://github.com/regent-ndongmo/UV_Projet.git