Abstract

My project consists of a web application for an architecture agency. It is composed of two parts.

The first part involves a website designed for prospective visitors:

This website allows them access any designs made by the agency. For example, photos showing these designs that can be either achievements or rehabilitation. We can read a description with each picture to allow the visitor to be informed on the work done. The visitor has the opportunity to contact the architecture agency through the completion of a form and can leave a message with his E-mail and phone number. The visitors who leave a message must identify themselves as individuals or as members of a company.

The second part includes a website designed for the administrator:

The administrator website is available for the agency's staff. The administrator is in charge of managing the database. The latter contains information about the clients and companies such as builders and suppliers for example.

The administrator has access to a web interface that allows him / her to add or modify a client. He also has the ability to add or remove some company's data in the database.

The administrator can view the messages left by the site visitors and reply to them.

I am planning in the future to create a mobile application linked to this website.

Résumé

Mon projet est une application web pour une agence d'architecte. Elle est compose de deux parties.

Une partie du site Web est conçu pour les visiteurs du site:

Ce site leur permet d'accéder à des projets réalisés par l'agence. Par exemple, une rubrique composée de photos permet de montrer les créations de l’agence, qui seront regroupés soit en réalisations soit en réhabilitations. Chaque image est accompagnée d’une courte description, pour permettre au visiteur d'être informé du travail effectué sur l’œuvre. Le visiteur a aussi la possibilité de contacter l'agence d'architecte à travers un formulaire et peut laisser un message accompagné de son E-mail et son numéro de téléphone. Ces visiteurs sont alors identifiés soit en tant que particulier ou soit en tant qu’entreprise.

Une autre partie du site Web concerne l'administrateur:

La partie administration n’est accessible que pour le personnel de l'agence. L'administrateur est responsable de la gestion de la base de données. Celle-ci est composée de tables qui contiennent des informations concernant les clients et les entreprises qui sont des entrepreneurs ou des fournisseurs. L'administrateur a accès à une interface web qui lui permet d'ajouter ou modifier un client. Il a également la possibilité d'ajouter ou de supprimer les données d'une entreprise dans la base de données. L'administrateur a la possibilité de visualiser les messages laissés par les visiteurs du site et y répondre.

Je prévois de créer une application mobile liée à ce site.

Introduction

Dans le cadre de ma formation de développeuse d’applications numériques, j’ai choisie de réaliser le site web d’une agence d’architecte comme projet personnel.

Ce site web est avant tout un outil de communication qui permet à l’agence d’être visible sur internet. De ce fait c’est un site dynamique dont le visiteur peut naviguer dans diverses pages web et communiquer avec l’agence par l’intermédiaire d’un formulaire de contact.

Une seconde partie "administrateur" permet à l’agence de gérer les données concernant les clients et les entreprises en contact avec l’agence.

L’objectif de ce projet est d’appliquer des compétences acquises au cours de cette formation et d’apprendre à gérer un projet professionnel en respectant les exigences du client grâce à une méthodologie de travail qui comprend des étapes de réalisation tels que :

- la présentation du projet et de l’analyse des besoins avec le client

- la conception avec les procédures qui assurent le développement web

- l’utilisation des technologies et des langages nécessaires à la réalisation du projet

- le développement proprement dit le codage et le design du site.

Je vais détailler les différentes étapes de mon travail en étant la plus précise possible et présenterai le résultat obtenu en accord avec les besoins du client. Pour finir, la dernière partie sera consacrée au bilan de ce projet par une analyse du travail effectué.

I - Présentation du projet

I.1 - Description

Le projet "Création d’un site web pour une Agence d’Architecte" a pour objectif la réalisation d’un site internet. Ce site permet de rendre visible l’agence sur des moteurs de recherche et de générer des contacts de futurs clients.

Le site s’adresse d’abord à des futurs propriétaires, mais aussi à des entreprises (type entrepreneur ou fournisseur) désireuses de collaborer avec l’agence. Le but de ce travail est de développer un site web dynamique qui permettra aux visiteurs d’être informés sur l’agence, de visionnés les projets réalisés et de contacter l’agence par l’intermédiaire d’un formulaire de contact.

Ce site est composé d’une seconde partie "administrateur" qui n’est accessible que par le personnel de l’agence après authentification. L’administrateur sera chargé de la gestion de la base de données. L’objectif est de facilité la maintenance de celle-ci même si la personne chargée de ce travail n’a aucune connaissance en programmation. Elle pourra ainsi ajouter, modifier ou supprimer des informations.

I.2 - Analyse des besoins

Les échanges avec l’Agence d’Architecte a permis de définir les différents objectifs du site. Le site s’adressant à de futurs clients et à des entreprises, l’agence veut un site web attractif (avec des images issues de réalisations) dont la première page d’accueil introduit l’agence et donne la possibilité au visiteur de naviguer vers d’autres pages tels que :

- *l’agence d’architecte* qui présente l’agence

- *projets* réalisés (créations et réhabilitations) avec des photos et des descriptions

- *habitat développement durable*

- *contact* formulaire avec nom, prénom, téléphone, email, et spécification si *particulier* ou *entreprise*.

La seconde partie "administrateur", permet à l’agence d’administrer une base de données pour stocker et gérer les informations concernant les clients, les entreprises, les projets réalisés et les travaux effectués, ainsi que les messages laissés sur le site. L’insertion des données dans la base ce fera par des formulaires :

- *ajouter un client* avec ces coordonnées

- *ajouter* *un* *projet* avec de nom projet, date, statut (neuf ou ancien), images …

- *ajouter une entreprise* avec ces coordonnées, numéro SIRET, personne à contacter…

- *ajouter une activité* avec sa description.

Par l’intermédiaire d’une page web, l’administrateur du site aura la possibilité d’agir sur les informations stockées dans la base autrement dit d’ajouter, de modifier et de supprimer les données concernant les clients, les projets, les entreprises et le type de travaux. Une autre page web, lui permet de visualiser et de supprimer les messages laissés par les visiteurs du site.

II - Conception graphique

La conception d’un site passe par une étape de modélisation en utilisant UML : Langage de Modélisation Unifié. C’est un langage visuel constitué d’un ensemble de schémas, appelés des diagrammes. Ces diagrammes vont modéliser le fonctionnement et les actions susceptibles d’être effectuées par l’application.

Pour modéliser les besoins de l’application, j’ai utilisé trois types de diagrammes :

- diagramme de cas d’utilisations

- diagramme de séquences

- diagramme d’activité.

II.1 - Diagramme de cas d’utilisations

Un diagramme de cas d’utilisations est utilisé pour donner une vision globale du comportement fonctionnel du système (application). Il représente un ensemble d’interactions (cas d’utilisations) entre un utilisateur (appelé *acteur*) et le système.

**Identification des acteurs :**

Les acteurs du système sont des entités externes du système qui interagissent avec lui. Dans cette application, il existe deux acteurs :

- le visiteur du site

- l’administrateur du site.

**Identification des cas d’utilisations :**

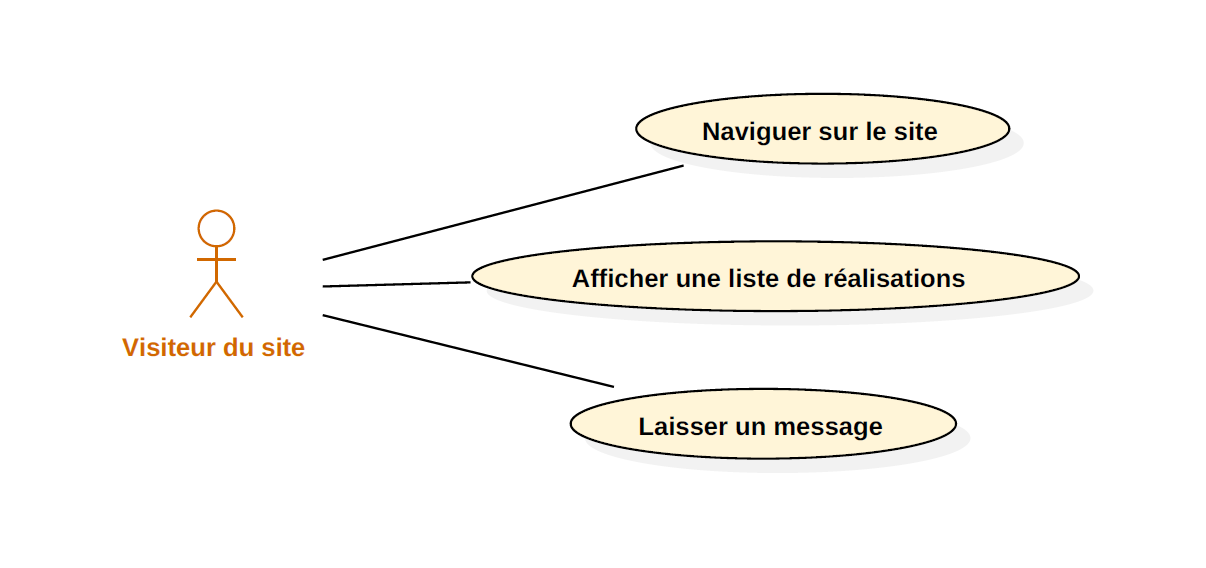
Un cas d’utilisation est utilisé pour définir le comportement du système tel que l’acteur le voit. Chaque cas d’utilisation spécifie une séquence d’actions que le système réalise en interagissant avec les acteurs. Les cas d’utilisations de cette application :

*Pour le visiteur :*

- naviguer sur le site : visualiser les autres pages web

- afficher des projets (réalisations) : visualiser des images

- laisser un message.



**Figure 1 :** Diagramme de cas d’utilisations pour un visiteur

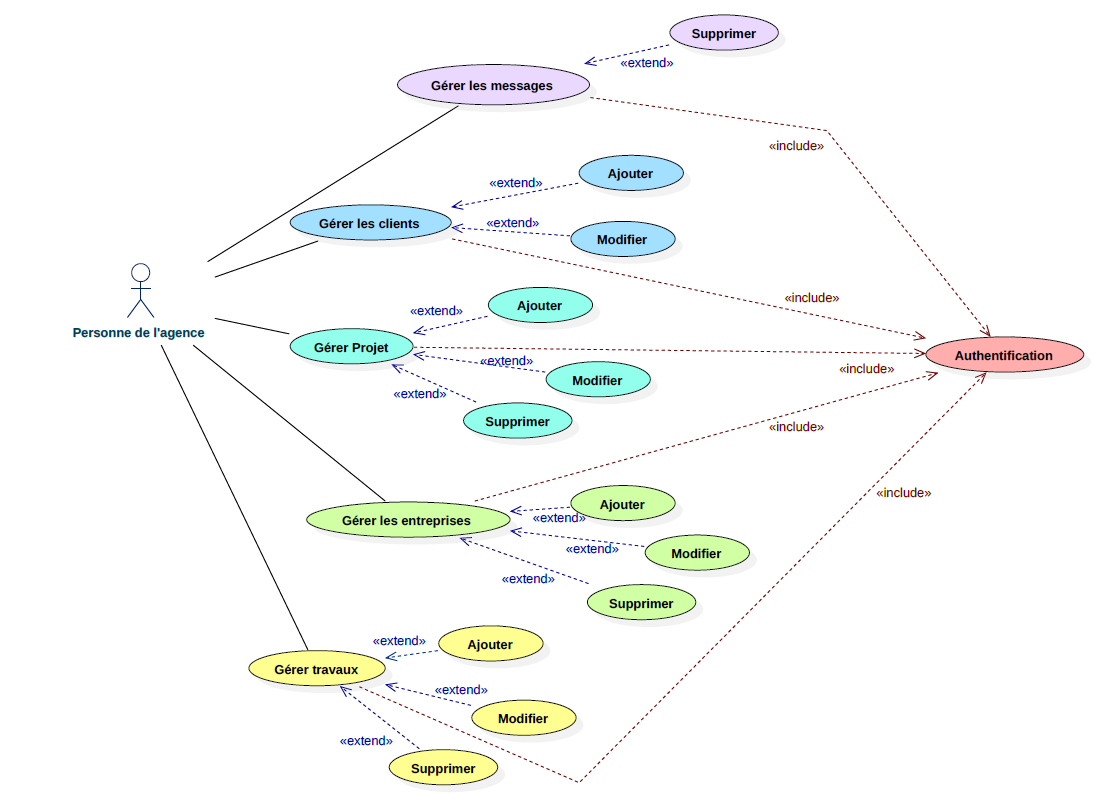
*Pour l’administrateur :*

- authentification : l’application vérifie que l’utilisateur (personne de l’agence) est bien ce qu’il prétend être et lui donne accès

- ajouter : pouvoir ajouter un client, un projet, une entreprise ou un type de travaux dans la base de données

- modifier : droit de modifier les informations dans la base de données

- supprimer : possibilité de supprimer les informations d’un projet, d’une entreprise ou d’un message.



**Figure 2 :** Diagramme de cas d’utilisations pour l’agence

II.2 - Diagramme de séquence

Un diagramme de séquences permet de décrire chaque cas d’utilisation en exposant en détail les opérations des interactions entre un acteur, le système et la Base de Données. Il est organisé en une séquence d’interactions dans le temps. Les objets impliqués dans l’opération sont répertoriés de gauche à droite en fonction du moment où ils prennent part dans la séquence.

Je vais présenter trois cas d’utilisation décliné en diagramme de séquence.

II.2.1 - Diagramme de séquence : "*Laisser un message par un visiteur*"

1 : un visiteur qui laisse un message doit demander le formulaire

2 : le système affiche le formulaire de contact

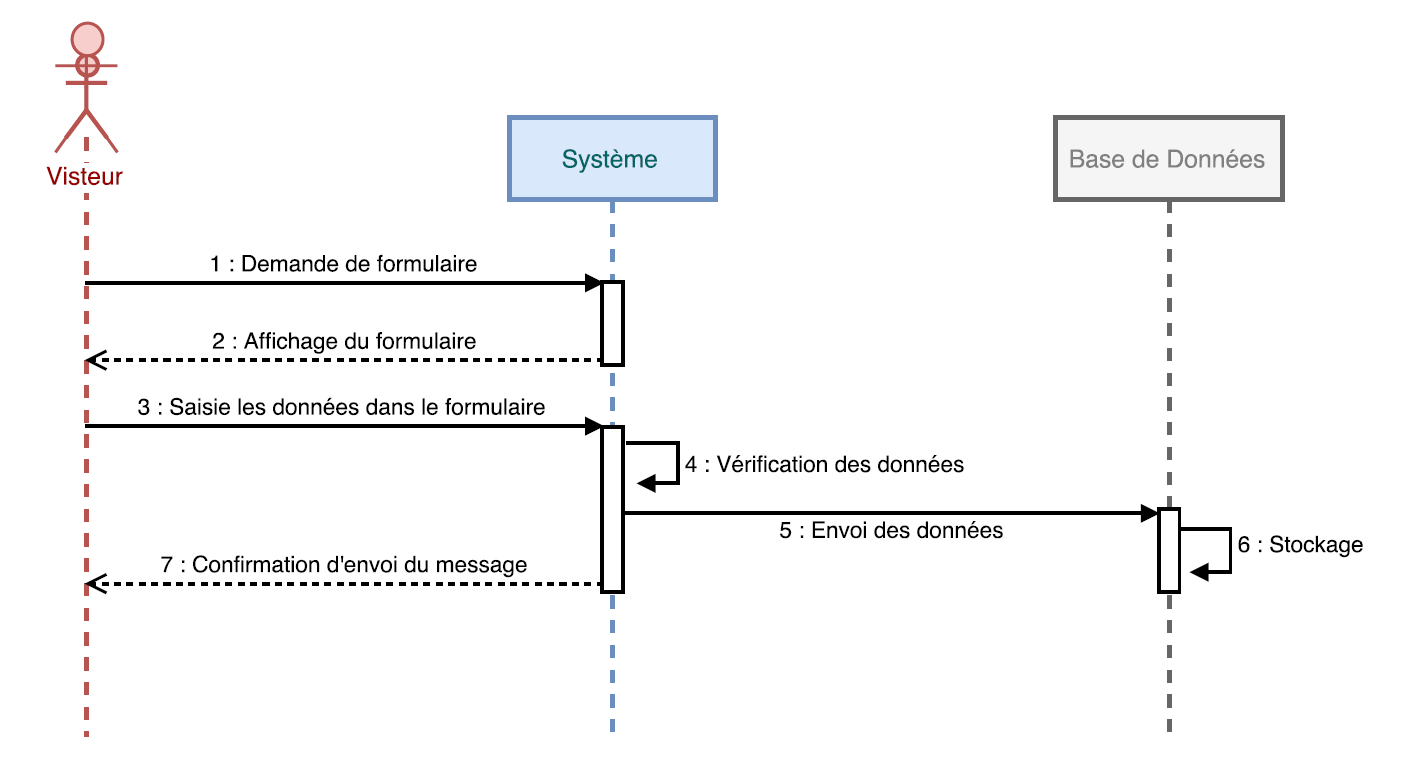
3 : le visiteur rempli le formulaire

4 : le système vérifie si les données sont saisies et si elles correspondent au format attendu

5 : le système envoi les données à la base de données

6 : la base de données stocke les nouvelles données

7 : le système confirme au visiteur l’envoi du message



**Figure 3 :** Diagramme de séquence pour *laisser un message*

II.2.2 - Diagramme de séquence : "*Authentification pour accéder à la partie administrateur*"

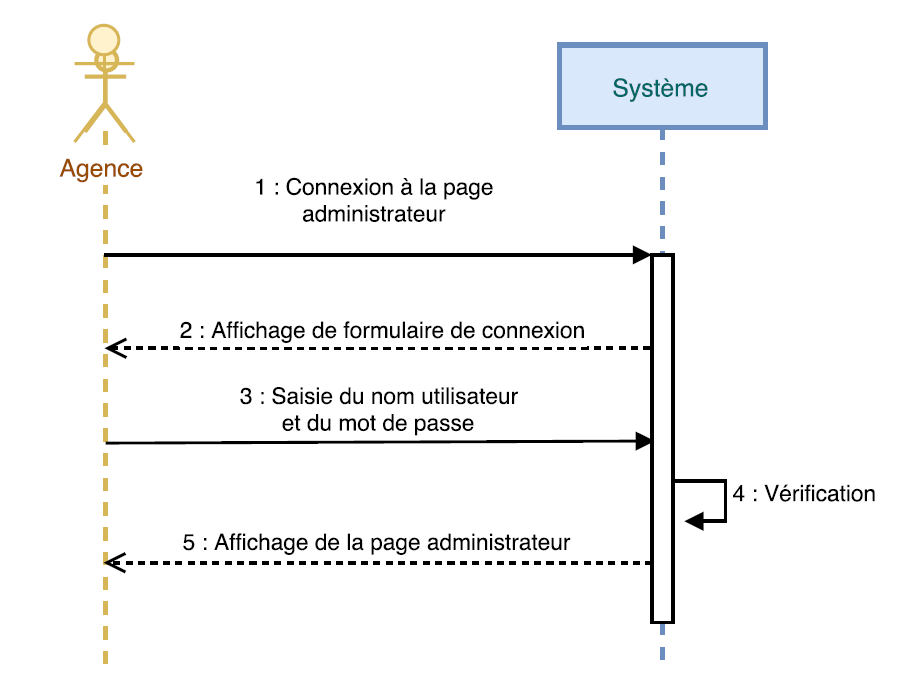
1 : l’administrateur (agence) ce connecte à la page administrateur

2 : le système affiche le formulaire de connexion

3 : l’administrateur saisie le nom d’utilisateur et le mot de passe

4 : le système vérifie les données saisies

5 : le système affiche la page administrateur



**Figure 4 :** Diagramme de séquence pour *authentification*

II.2.3 – Diagramme de séquence : "*Modification des données d’un client*"

1 : l’administrateur demande le formulaire client

2 : le système envoi des données client à la base de données

3 : la base de données recherche les données client

4 : la base de données recharge les données client dans le formulaire

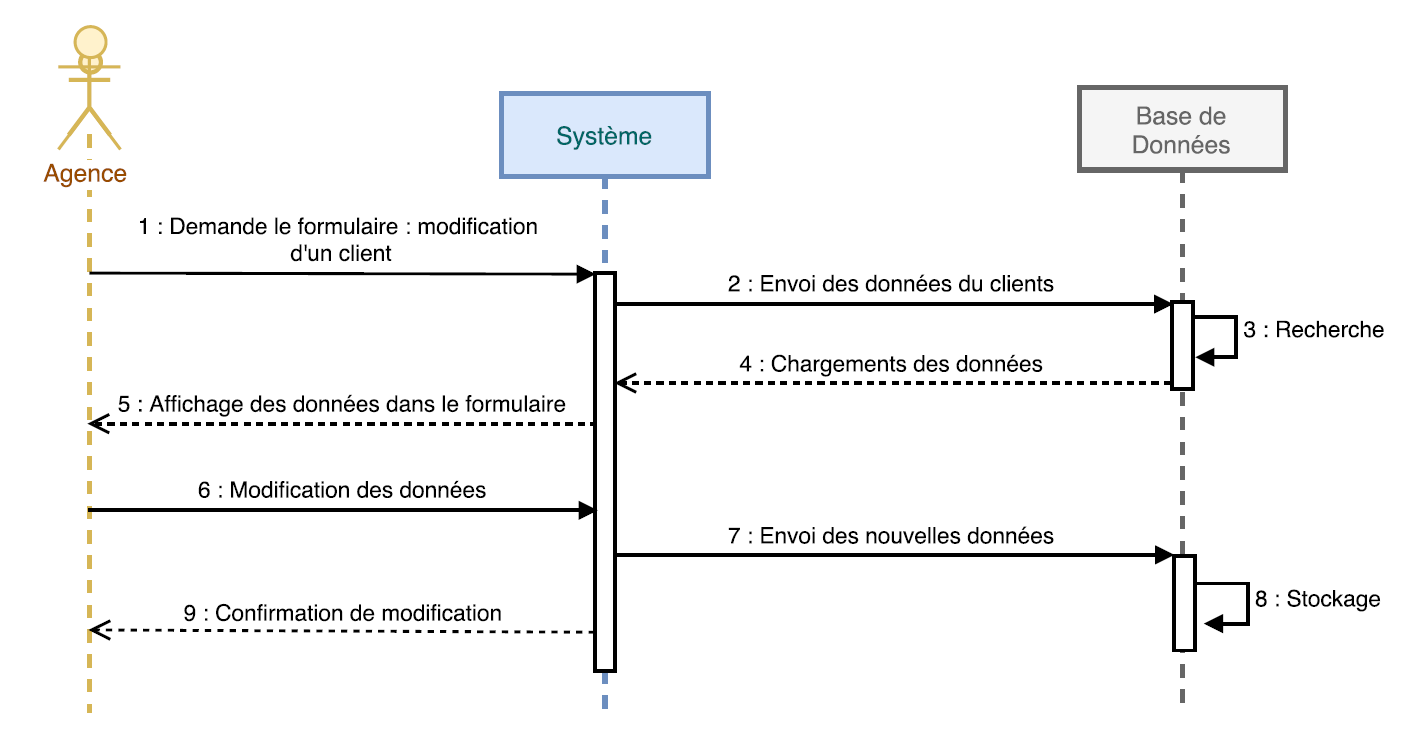
5 : le système affiche le formulaire avec les données

6 : l’administrateur modifie les données désiré

7 : le système envoi les nouvelles données à la base de données

8 : la base de données stocke nouvelles données

9 : le système confirme la modification des données



**Figure 5 :** Diagramme de séquence *modification de données*

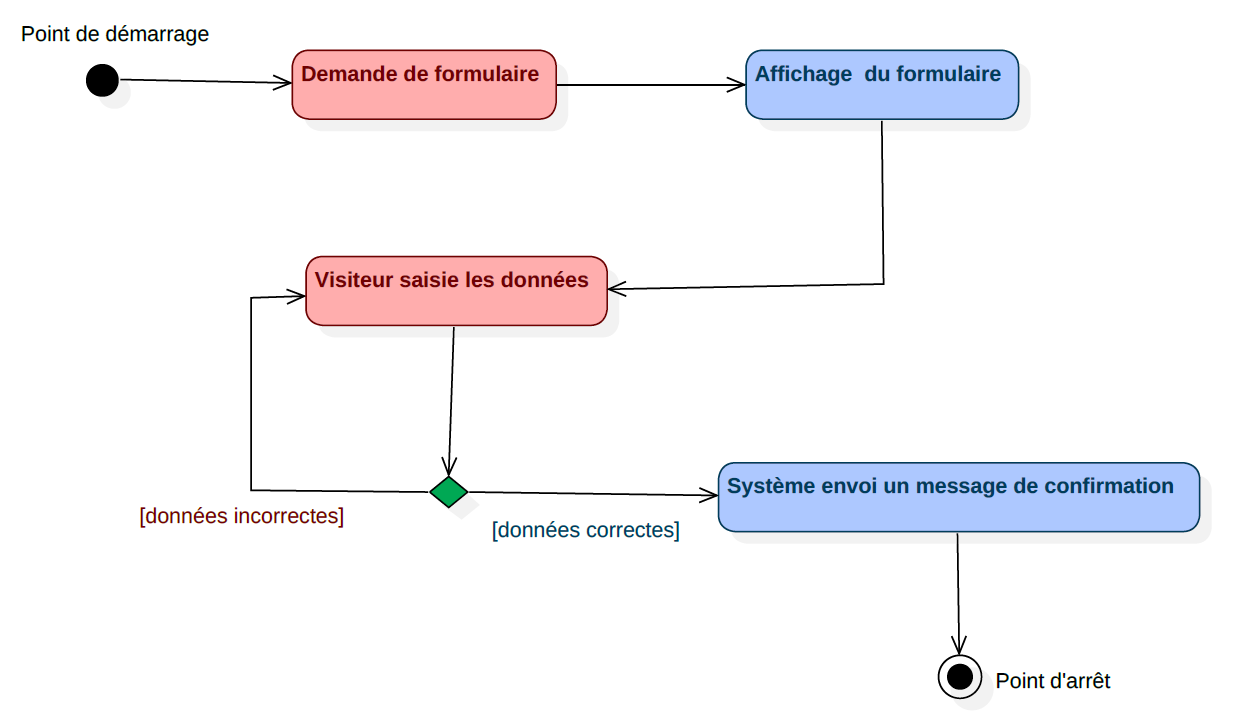
II.3 - Diagramme d’activité

Un cas d’utilisation peut contenir plusieurs scénarios alternatif et rend difficile la vision de l’ensemble des actions. Le diagramme d’activité est un moyen graphique de représenter cette vision d’ensemble.

Je vais reprendre les mêmes trois cas d’utilisation.

II.3.1 - Diagramme d’activité : "*Laisser un message par un visiteur*"

Lors de la demande du formulaire par le visiteur, le système lui affiche le formulaire, après saisie des données par le visiteur, le système vérifie la validité des données et envoie un message de confirmation de l’envoi du message (si données correctes) sinon il affiche un message pour modifier les données (si données incorrectes).

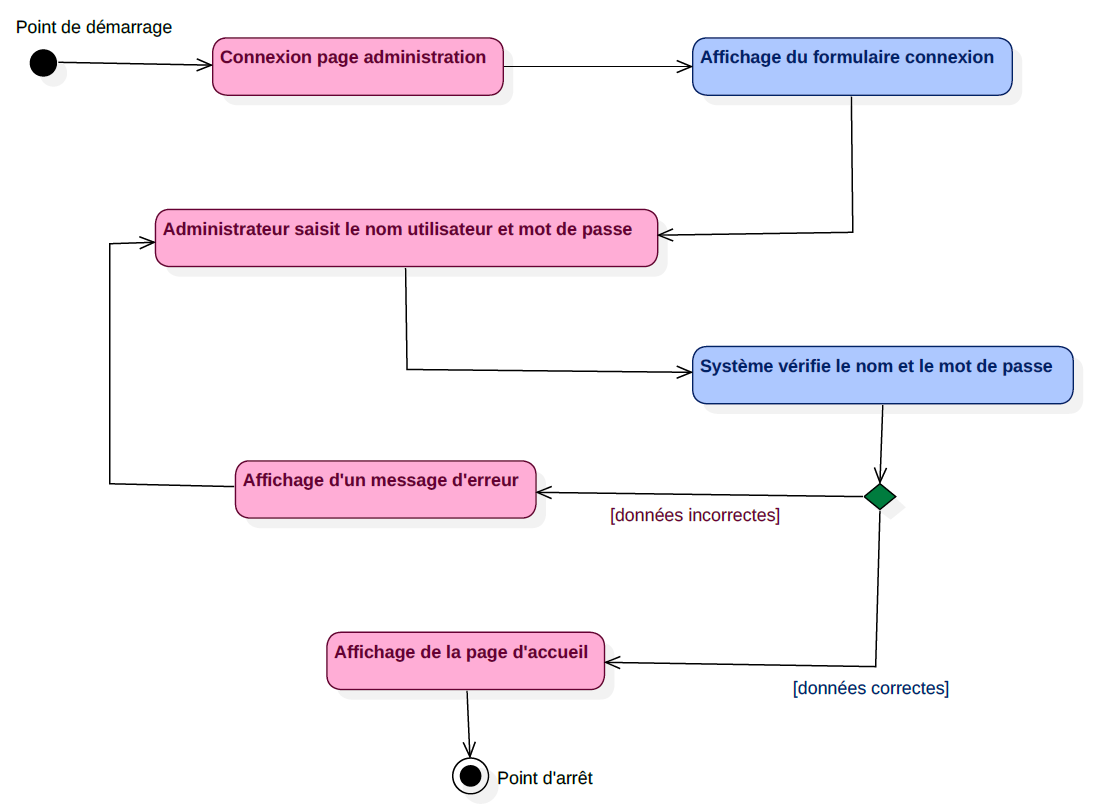


Point de décision

**Figure 6 :** Diagramme d’activité *laisser un message*

II.3.2 - Diagramme d’activité : "*Authentification par l’administrateur"*

La connexion à la page administration par l’agence conduit le système à afficher le formulaire de connexion, après saisie du nom d’utilisateur et du mot de passe par l’administrateur, le système vérifie la validité des données et affiche la page d’accueil (si données correctes) sinon affiche un message d’erreur et demande de modifier les données (si données incorrectes).

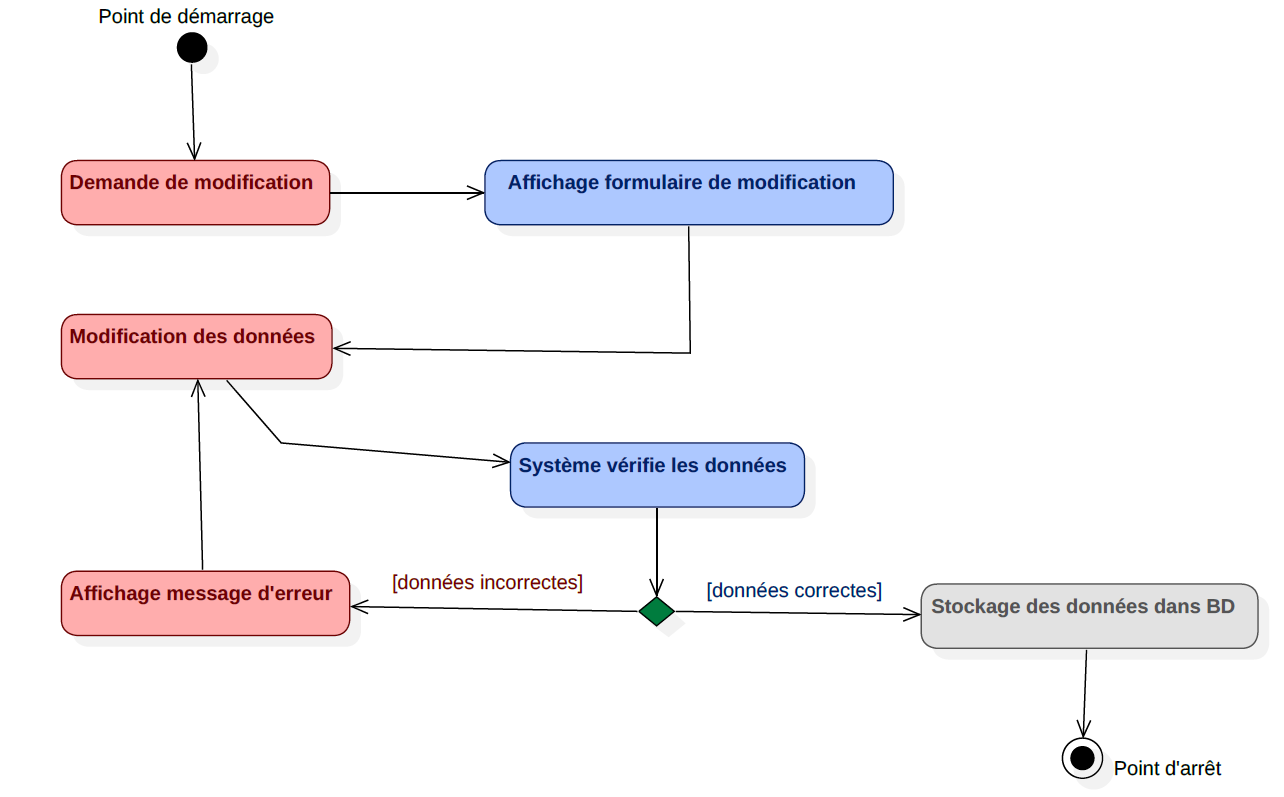


Point de décision

**Figure 7 :** Diagramme d’activité *authentification*

II.3.3 - Diagramme d’activité : "*Modification de données par l’administrateur"*

Après demande de modification de données par l’administrateur, le système affiche le formulaire à modifier avec les données, puis l’administrateur modifie les données, le système vérifie les données modifiées et les envois en base de données (si données correctes) sinon affiche un message d’erreur pour modifier les données (si données incorrectes).



Point de décision

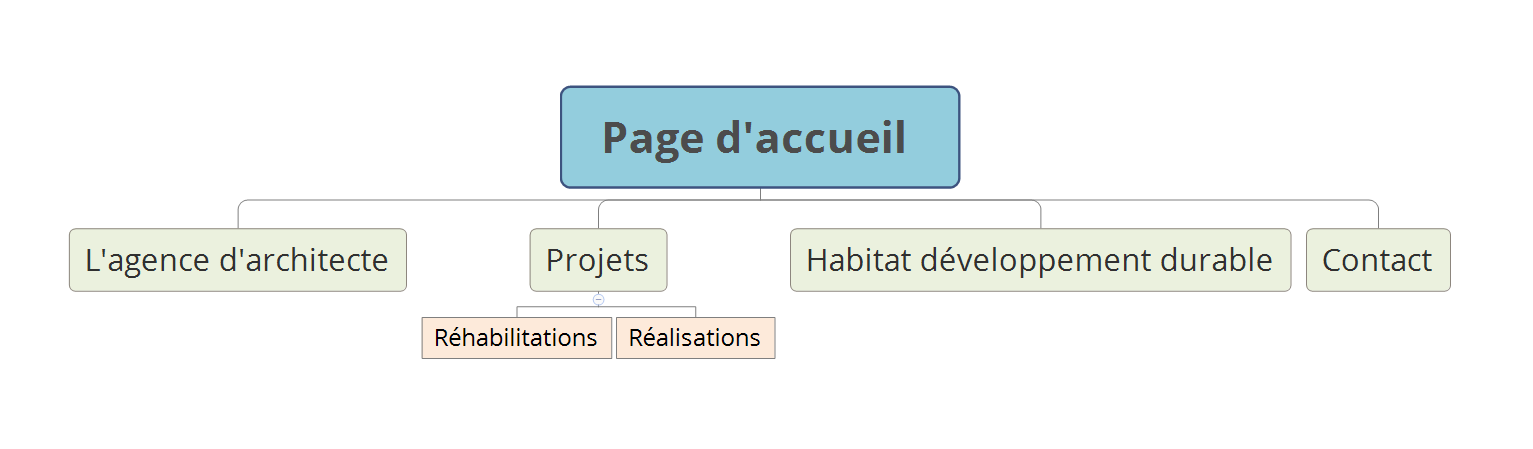
**Figure 8 :** Diagramme d’activité *modification*

II.4 - Arborescence du site

L’arborescence du site permet de définir la structure globale (où l’architecture) du site en s’appuyant sur les cas d’utilisations attendus par l’application. Elle détermine l’organisation des pages en rubriques et définit les différents niveaux de navigation.

II.4.1 - Partie visiteur

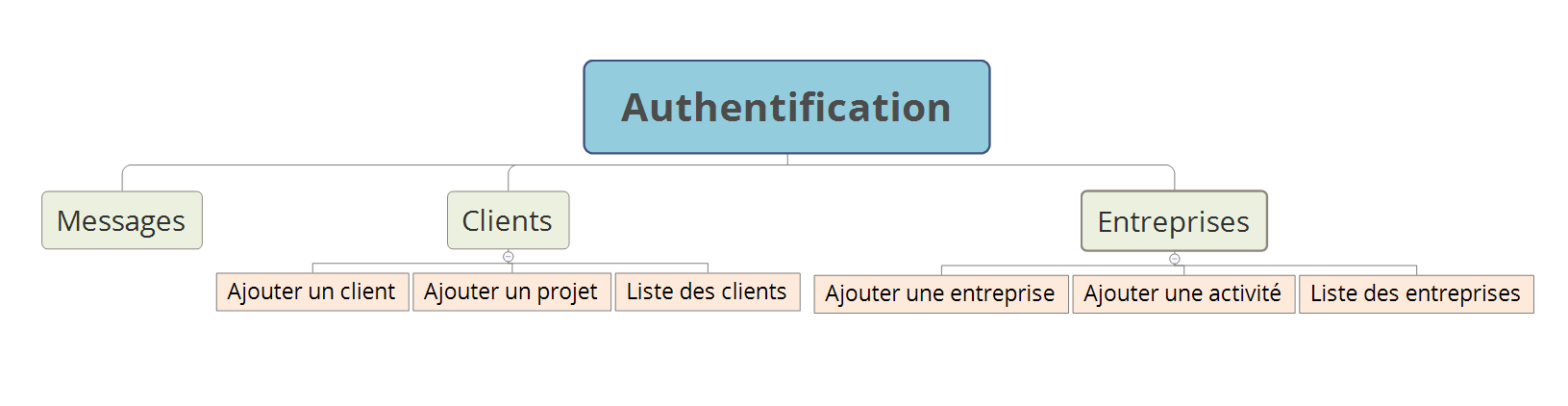
L’arborescence du site partie visiteur est constitué d’une page d’accueil qui permet d’accéder directement à quatre nouvelles pages, dont la page *Projets* donnent accès à d’autres pages.



**Figure 9 :** Arborescence partie *Visiteur*

II.4.1 - Partie administrateur

L’accès à la partie administrateur du site nécessite une authentification. Après authentification on accède à la page des messages laissés par les visiteurs du site, qui est composée d’un menu déroulant permettant d’accéder à des formulaires pour ajouter, modifier ou supprimer les données enregistrés dans la base de données.



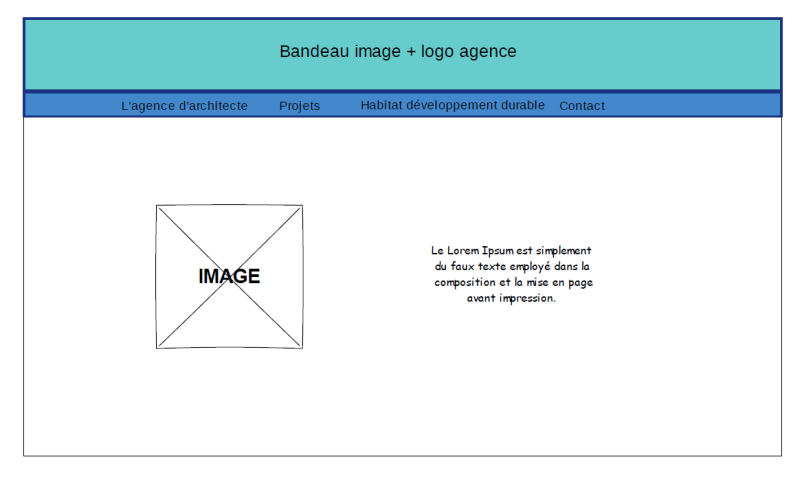
**Figure 10 :** Arborescence partie *Administrateur*

II.5 - Maquettage

Le maquettage d’un site web c’est un dessin, réalisé avec un logiciel dédié, qui permet de déterminer la mise en page et l’organisation des différents éléments qui vont composer chaque page du site. Cette maquette donne ainsi un aspect visuel du futur site internet.

Je vais présenter quatre pages qui composent les deux parties du site.

II.5.1 - Partie Visiteur



**Figure 11 :** Maquette de la page *Accueil*

Constituée d’un diaporama de photos des

réalisations et d’un texte d’introduction.



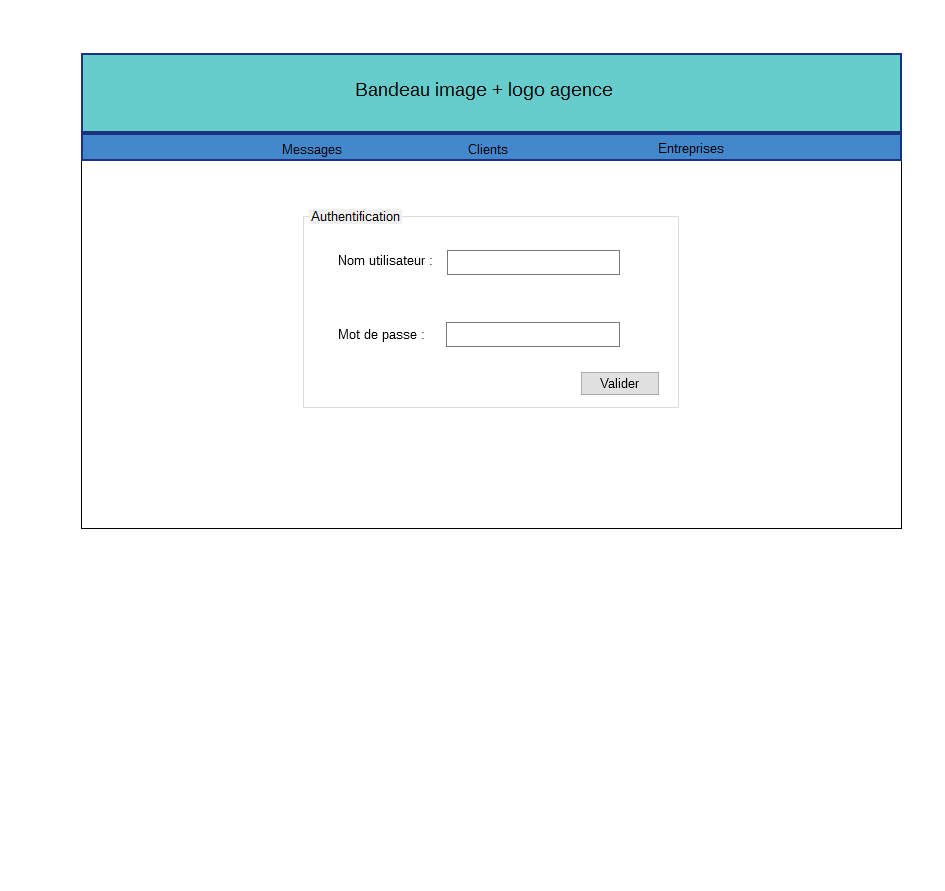
**Figure 12 :** Maquette de la page *Contact*

Composée d’un formulaire de contact,

des coordonnées de l’agence d’architecte

et d’un Google map.

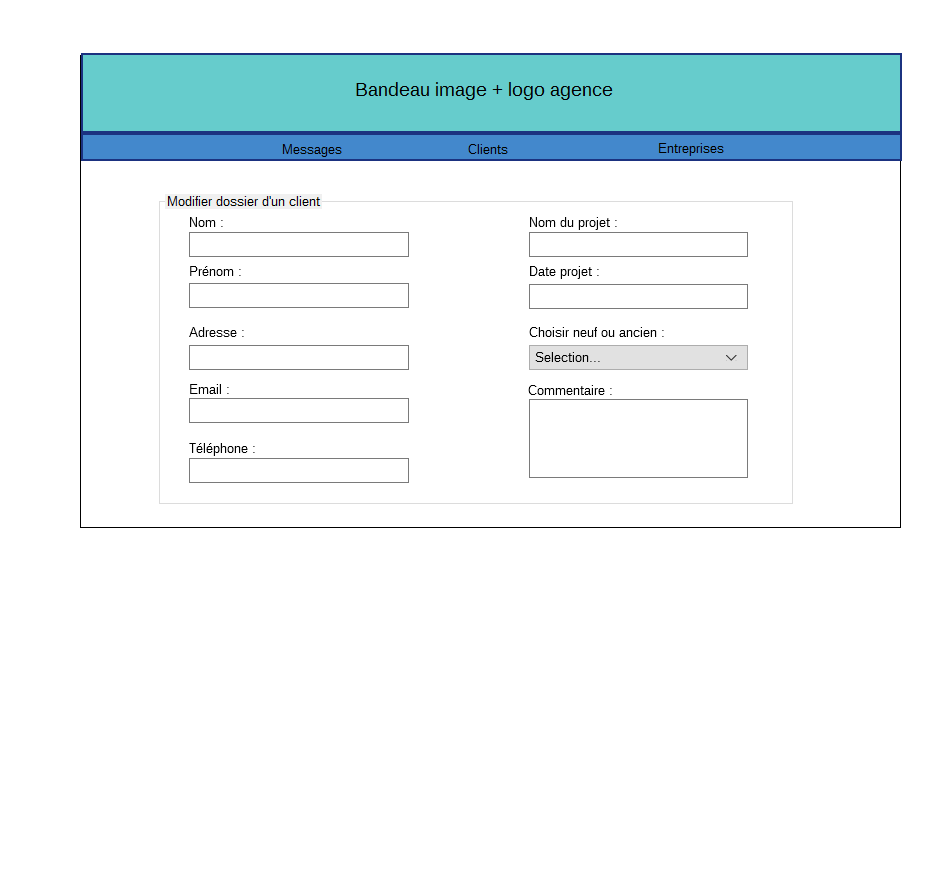
II.5.2 - Partie Administrateur



**Figure 13 :** Maquette de la page *Authentification*

Authentification avec nom d’utilisateur et

mot de passe.



**Figure 14 :** Maquette de la page *Modifier*

Possibilité de modification d’un dossier client

et de son projet.

III - Conception de la Base de Données

Une base de données est un système qui permet d’enregistrer des données de façon organisée et hiérarchisée. Ces données sont accessibles par des applications et par des utilisateurs pour une consultation, une saisie ou une mise à jour, tout en s’assurant des droits accordés à ces derniers. L’avantage de l’utilisation des bases de données est de pouvoir être accessible par plusieurs utilisateurs simultanément.

Pour pouvoir contrôler les données et les utilisateurs, il existe un Système de Gestion de Base de Données (SGBD) qui permet de gérer l'accès aux données de façon simple, d’autoriser un accès aux informations à de multiples utilisateurs et de manipuler les données présentes dans la base de données (consultation, insertion, suppression, modification).

Le langage utilisé pour manipuler les bases de données et le SQL. Il permet :

- La maintenance des tables : création, suppression, modification de la structure des tables

- La manipulation des données : sélection, modification, suppression d'enregistrements

- La gestion des droits d'accès aux tables : contrôle des données, droits d'accès, validation des modifications.

Pour concevoir cette base de données je me suis appuyé sur la méthode Merise, basée sur le principe de séparation des données et des traitements. Elle propose des modèles qui sont répartis sur trois niveaux :

- conceptuel

- logique

- physique

III.1 - MCD : Modèle Conceptuel des Données

Le Modèle Conceptuel des Données est une représentation graphique et structurée des informations mémorisées par un Système d’Information. Il est basé sur les notions d’**entités** et d’**associations**. D’où le terme de modèle **entités - associations**.

III.1.1 - Les entités

Une entité est unique et est décrite par un ensemble de propriétés appelées *attributs*. Chaque entité possède un attribut qu’on appelle "identifiant" qui est unique et souvent de type entier qui s’incrémente pour chaque occurrence.

III.1.2 - Les associations

Une association définit un lien sémantique entre une ou plusieurs entités. Elle permet de préciser la signification vis-à-vis des entités qu’elle relie. On désigne en général les associations par des noms de verbe à l’infinitif.

III.1.3 - Les cardinalités

La cardinalité caractérise la participation d’une entité à une association. Elle représente le nombre de participation d’une entité à une association.

Donc, pour une association de 2 entités, il y a quatre cardinalités à indiquer (deux de chaque côté). Il existe trois valeurs types : 0, 1 et n (plusieurs).

*Exemple Association entre Clients et Projets que j’ai développée dans mon projet:*

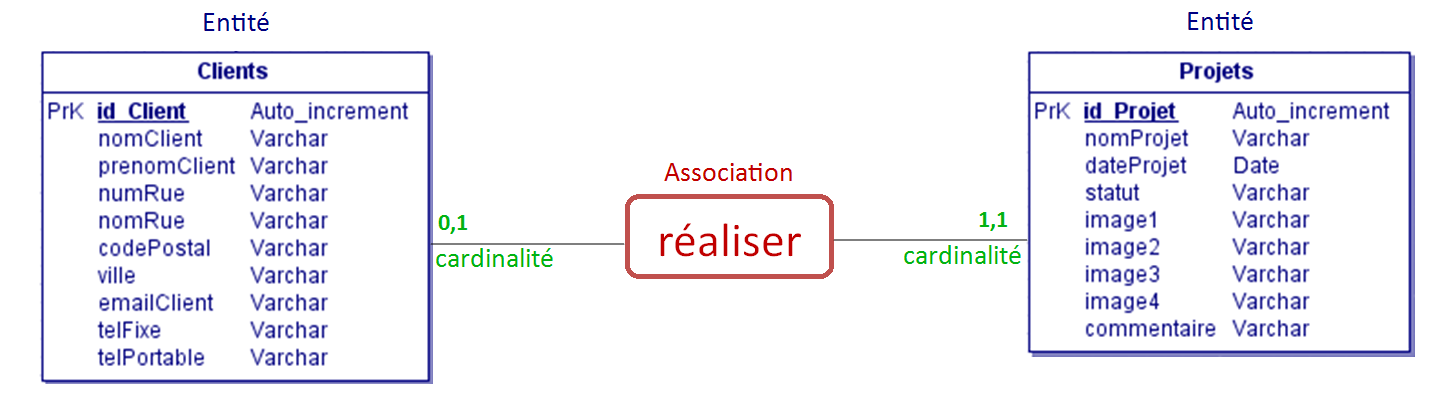
- entités : Clients et Projets

- association : réaliser

- cardinalité : - 1 client réalise 0 ou plusieurs projets

- 0 ou plusieurs projets sont réalisés pour 1 client

Identifiant



**0,n**

Attributs

**Figure 15 :** Exemple du modèle Entités - Association

III.1.4 - Dictionnaire des Données

Le dictionnaire des données est un tableau qui recense et décrit toutes les entités, et leurs attributs, du modèle MCD à partir des données recueillies.

Le dictionnaire des données du site de l’agence d’architecte comprend les colonnes suivantes :

- Nom des entités

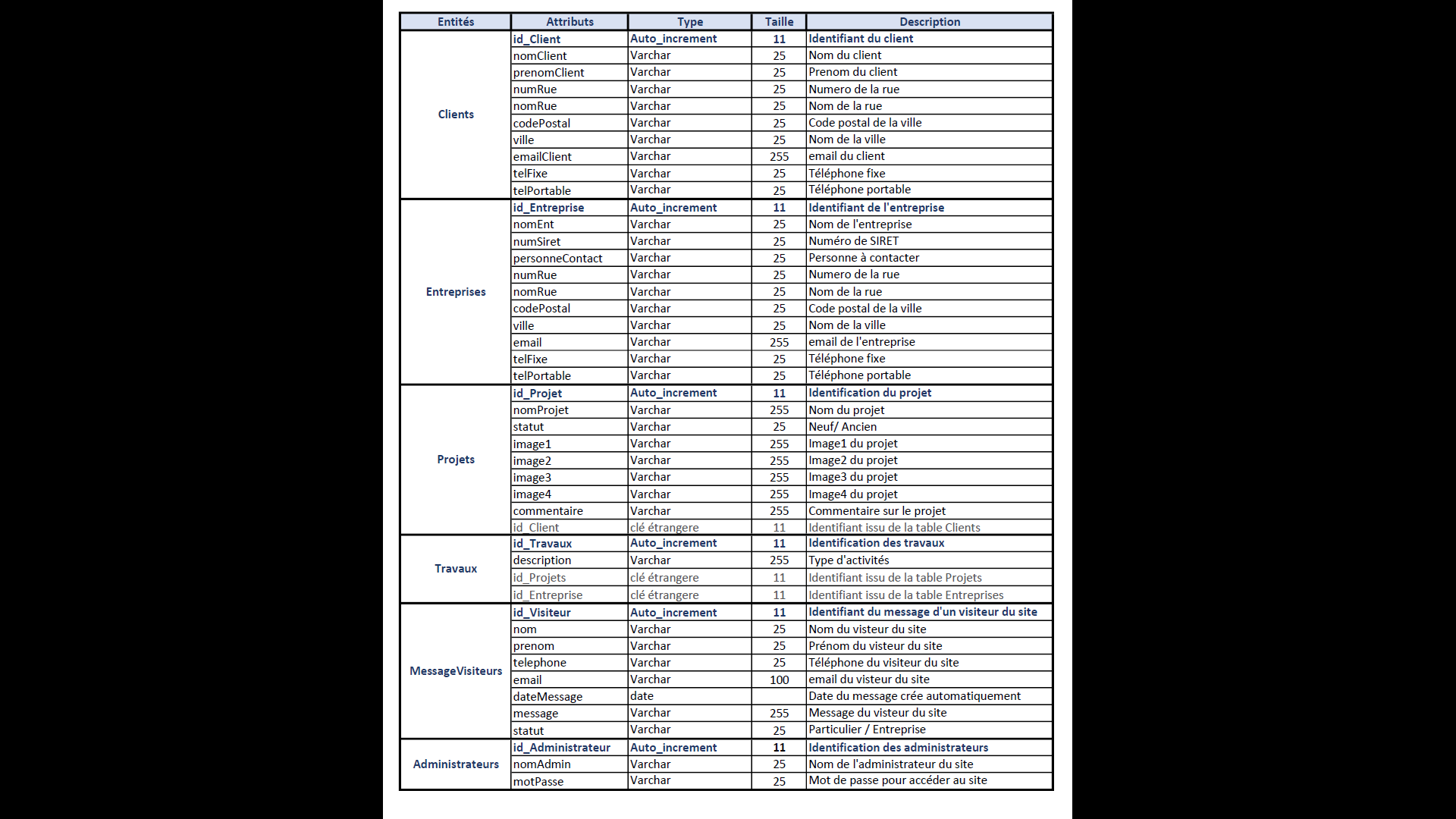
- Nom des attributs

- Type des attributs

- La taille des attributs

- Description des attributs

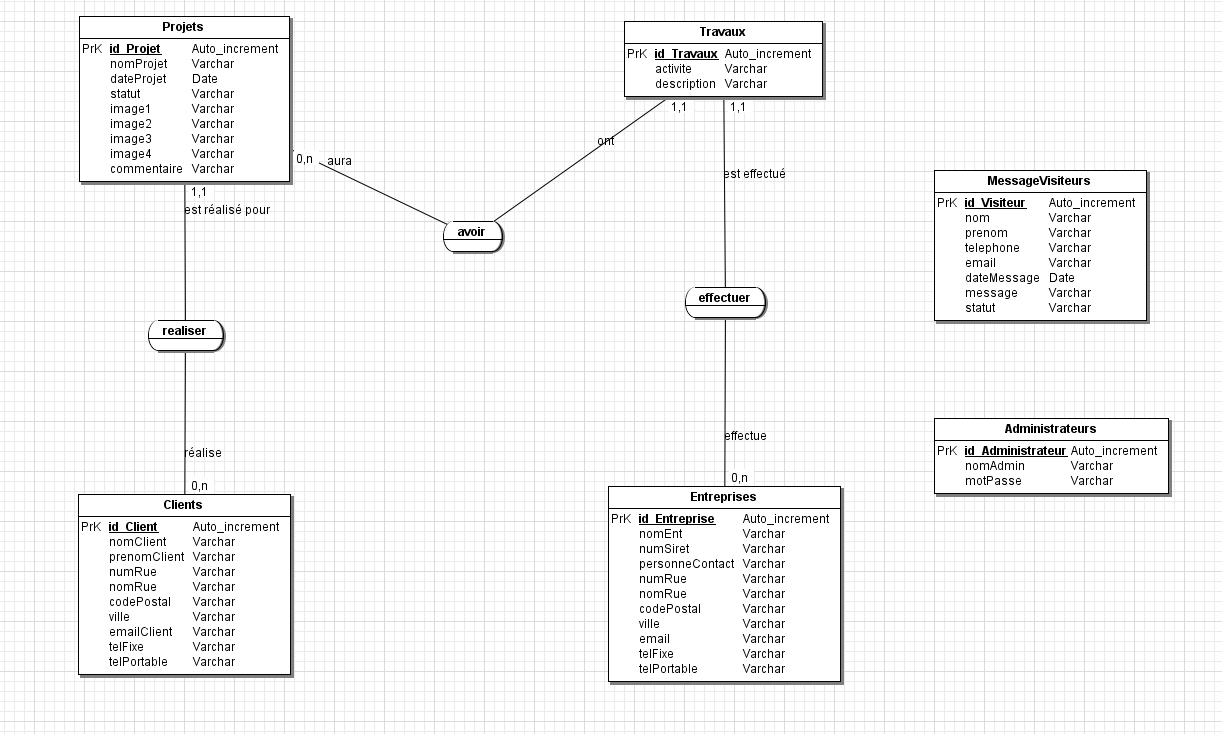
Toutes les données du dictionnaire des données figurent dans le MCD.



**Figure 16 :** Dictionnaire des Données

III.1.5 - Elaboration du MCD

Avec toutes ces connaissances, il est donc possible d’élaborer le MCD complet à partir des données présentes dans le dictionnaire des données :



**Figure 17 :** MCD du site de l’agence d’architecte

**Remarque :** les entités *MessageVisiteurs* et *Administrateurs* sont nécessaires pour stocker les données issues du remplissage du formulaire par les visiteurs et pour stocker les données des administrateurs respectivement.

III.2 - MLD : Modèle Logique des Données

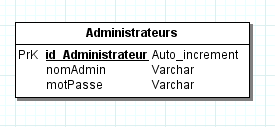
Après la modélisation conceptuelle, la méthode Merise fournit un procédé permettant d’aboutir à la structure finale de la base de données relationnelle. C’est le Modèle Logique des Données qui prépare l’implémentation des données au niveau physique. Le modèle logique dans une base de données relationnelle transforme les entités en table et détermine pour celles-ci la clé primaire. L’observation des cardinalités de chaque association permet de savoir si une clé étrangère doit être ajoutée dans une table ou si l’association nécessite la création d’une nouvelle table.

Le passage du MCD au MLD ce fait grâce à des règles de conversion :

III.2.1 - Règle 1

Toute **entité** devient une **table** dans laquelle les **attributs** deviennent les **colonnes** de la table. L’**identifiant** de l’entité constitue alors la **clé primaire** de la table.

table



clé primaire

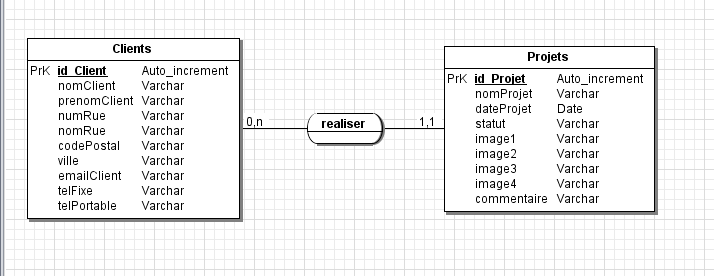
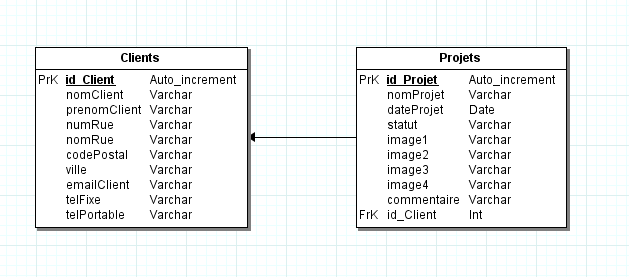
attribut

**Figure 18 :** MLD de la table Administrateurs

III.2.2 - Règle 2

Toute **association binaire** de type : 1,1 - 1,n ou 1,1 - 0,n se traduit par la création d’une **clé étrangère** dans l’association correspondant à l’entité côté 1,1. Cette clé étrangère référence la clé primaire correspondant à l’entité côté 1,n ou 0,n.

clé primaire



clé étrangère

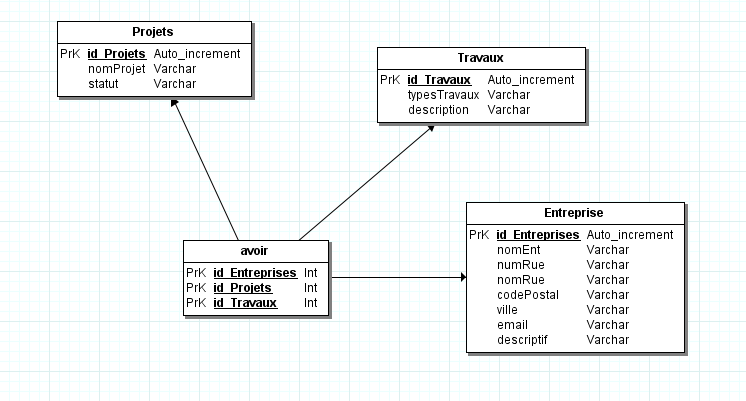
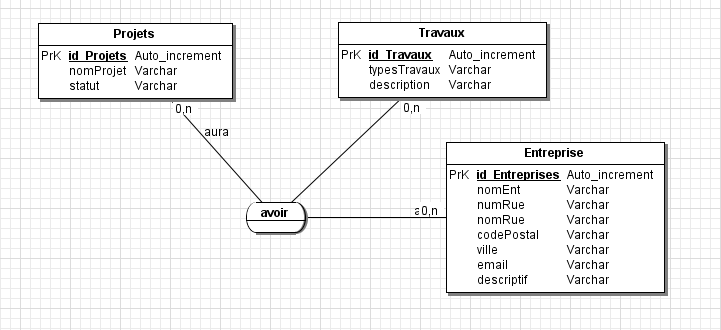
**MCD**

**MLD**

**Figure 19 :** MLD : association binaire de type 1,1 - 0,n

III.2.3 - Règle 3

Toute **association N-aires**, porteuse ou non d’attributs, ayant les cardinalités 0,n ou 1,n se transforme en une table ayant comme clé composite les colonnes issues des identifiants des entités participant à l’association



**MLD**

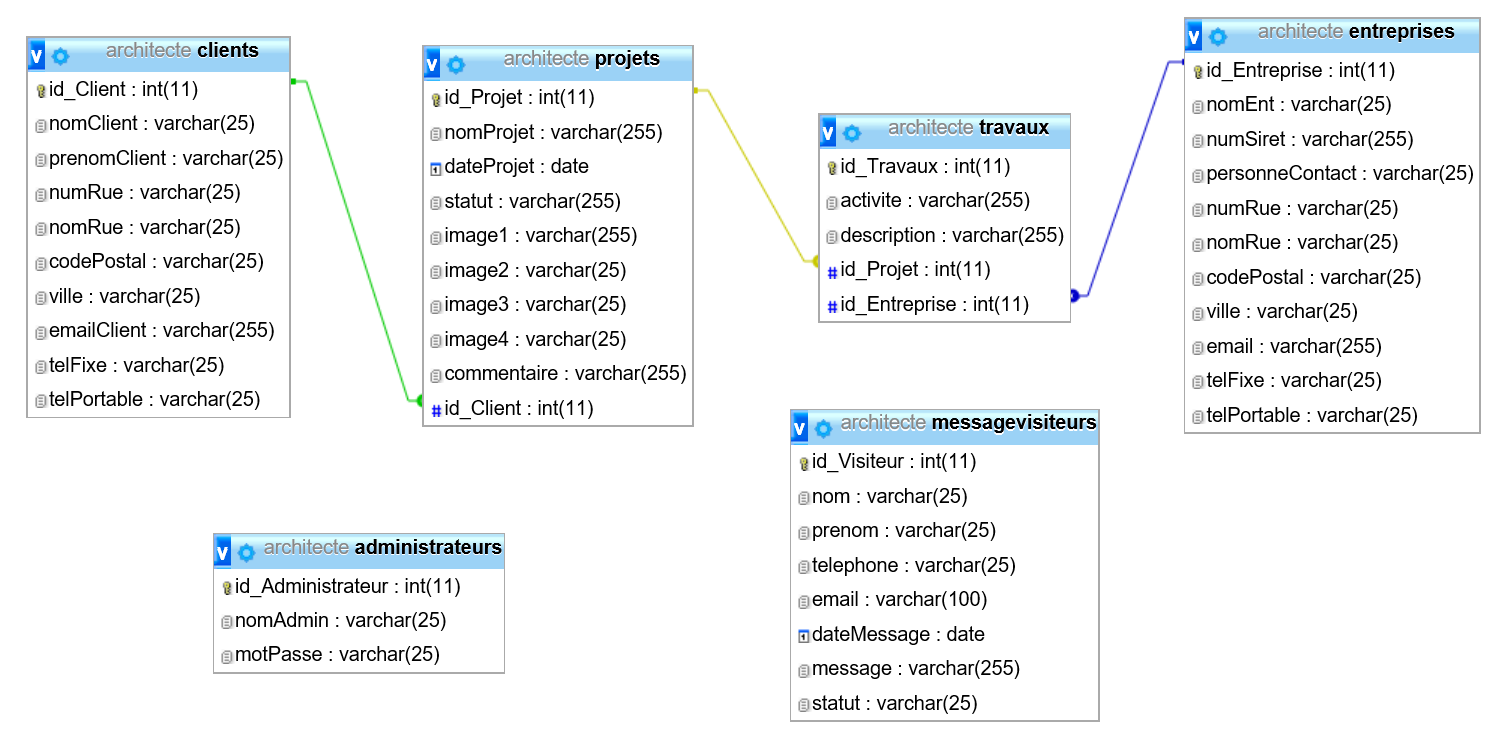
**MCD**

clé primaire composite

**Figure 20 :** MLD : association N-aire de type 0,n - 0,n

III.2.4 - Elaboration du MLD

Avec ces différentes règles de conversion, il est possible de convertir le MCD en MLD :

- *sous forme de tables*

**Figure 21 :** MLD du site de l’agence d’architecte

- *sous forme de Modèle Relationnel des Données : MRD*

**Clients** (id\_Client,nomClient,prenomClient,numRue,nomRue,codePostal,ville,emailClient,telFixe,telPortable)

**Projets** (id\_Projet,nomProjet,dateProjet,dateProjet,statut,image1,image2,image3,image4,commentaire,id\_Client#)

**Entreprises** (id\_Entreprise,nomEnt,numSiret,personneContact,numRue,nomRue,codePostal,ville,email,telFixe,telPortable)

**Travaux** (id\_Travaux,activité,description, id\_Projet#,id\_Entreprise#)

**MessageVisiteurs** (id\_Visiteur,nom,prenom,telephone,email,dateMessage,message,statut)

**Administrateurs** (id\_Administrateur,nomAdmin,motPasse)

clé étrangère

clé primaire

**table**

III.3 - MPD : Modèle Physique des Données

Dans la méthode Merise, le Modèle Physique des Données consiste à implémenter une base de données dans un SGBD (Système de Gestion de Base de Données). Le langage utilisé est le SQL.

La structure en tables et colonnes du MRD est conservé, mais on y ajoute les types de données de chacune des colonnes présents dans le dictionnaire des données.

L’**exemple** de script SQL pour la création du MPD des tables *Clients* et *Projets* est le suivant :

Table Clients Table Projets

|  |  |
| --- | --- |
| CREATE TABLE Clients(  id\_Client int (11) Auto\_increment NOT NULL,  nomClient Varchar (25) ,  prenomClient Varchar (25) ,  numRue Varchar (25) ,  nomRue Varchar (25) ,  codePostal Varchar (25) ,  ville Varchar (25) ,  emailClient Varchar (255) ,  telFixe Varchar (25) ,  telPortable Varchar (25) ,  PRIMARY KEY (id\_Client )  ); | CREATE TABLE Projets(  id\_Projet int (11) Auto\_increment NOT NULL,  nomProjet Varchar (255) ,  dateProjet Date ,  statut Varchar (255) ,  image1 Varchar (255) ,  image2 Varchar (25) ,  image3 Varchar (25) ,  image4 Varchar (25) ,  commentaire Varchar (255) ,  id\_Client Int,  FOREIGN KEY ( id\_Client)  REFERENCES Clients(id\_Client),  PRIMARY KEY (id\_Projet )  ); |

IV - Les outils utilisés

La création du site pour l’agence d’architecte a nécessité l’utilisation de différents logiciels pour la conception graphique, la conception de la base de données et le fonctionnement en local de l’application.

L’implémentation a été effectuée sous un environnement Windows et a demandé l’utilisation d’un éditeur de code, de divers langages et d’un framework.

IV.1 - Conception graphique

IV.1.1 - StarUML

**StarUML** est un logiciel de modélisation UML. Il m’a permis de créer les diagrammes :

- Diagramme de cas d’utilisation

- Diagramme d’activité

IV.1.2 - Draw.io

**Draw.io** est une application gratuite en ligne, accessible via un navigateur web, qui m’a permis de dessiner les diagrammes de séquences.

IV.1.3 - XMind

**XMind** est un logiciel de création d’organigrammes, qui m’a permis de créer les arborescences du site avec l’organisation des pages en rubriques et avec les différents niveaux de navigation.

IV.1.4 - Pencil

**Pencil** est un logiciel de réalisation de maquettes typographiques, qui m’a permis de concevoir mes interfaces graphiques avec la représentation d’éléments tels que : la barre de menus, boutons, images, etc.

 IV.2 - Conception de la Base des Données

**JMerise** est un logiciel dédié à la modélisation conceptuelle des données (MCD), basé sur la méthode Merise. Ce logiciel m’a permis d’obtenir le MCD et de déduire le MLD et le MPD afin de créer la base de données du site.

 IV.3 - WampServer

 **WampServer** est une plateforme de développement web sous Windows, permettant de faire fonctionner localement des scripts PHP. Cette plateforme comprend :

- un serveur http **Apache**

- un SGBD **MySql**

- un outil de gestion des SGBD **phpMyAdmin**

Grâce à WampServer j’ai testé mon site localement sans être obliger d’avoir un hébergement et/ou un accès à internet.

IV.5 - Visual Studio Code

L’éditeur que j’ai choisie pour génère du code est **Visual Studio Code**. C’est un logiciel qui permet d’éditer un site web pour Windows. Il intègre des fonctionnalités telles que la coloration syntaxique, indentation, auto complétion et supporte de nombreux langages.

IV.6 - Langages

IV.6.1 - HTML 5

Le **HTML5** (*HyperText Markup Language 5*) est un langage dit de "balise" ou de "structuration" permettant la conception de pages web avec des balises de formatage. Les balises permettent d’indiquer la façon dont doivent être présenté le document et les liens qu'il établit avec d'autres documents.

IV.6.2 - CSS 3

Le **CSS 3** (*Cascading Style Sheets* : feuilles de style en cascade) est un langage informatique complétant le HTML. Alors que le HTML structure la page Web, le CSS va la mettre en forme en y apportant du style.

IV.6.3 - PHP 7

Le **PHP** (*Hypertext Preprocessor)*, est un langage interprété (où langage de script) exécuté du côté serveur. Il est spécialement conçu pour le développement de pages web dynamiques.

Ses principaux atouts en font un des langages web le plus utilisé :

- Une grande communauté de développeurs partageant des centaines de milliers d'exemples de script PHP

- La simplicité d'écriture de scripts

- La possibilité d'inclure le script PHP au sein d'une page HTML

- La facilité d’accès aux Bases de Données (de nombreux SGBD comme MySQL) et pour le traitement des formulaires.

IV.6.4 - jQuery

**jQuery** est une bibliothèque de JavaScript, qui vient compléter le HTML et le CSS afin de déterminer le contenu et l'apparence d’une page web, en y ajoutant une couche d'interactivité.

Elle facilite le développement rapide d’applications web en simplifiant de nombreuses tâches par la réduction de la quantité de code nécessaire.

Par ailleurs, jQuery utilise des sélecteurs d’éléments CSS, permettant d’être compatible avec la majorité des navigateurs.

Afficher l'image d'origineIV.6.5 - SQL

Le **SQL** (*Structured Query Language*) est un langage servant à exploiter des bases de données relationnelles.

Il permet par l’intermédiaire de requêtes :

- La création de base de données et des tables.

- L'ajout d'enregistrements sous forme de lignes.

- L'interrogation de la base.

- La mise à jour.

- Le changement de structure de la table: ajout, suppression de colonnes.

- La gestion de droits d'utilisateurs de la base.

****IV.7 - Bootstrap

**Bootstrap** est un framework front-end composé d'outils utiles à la création d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript.

V - Réalisation du site

La réalisation du site a nécessité que je réfléchisse à l’organisation des éléments qui constituent une page par une mise en forme effectuée principalement par HTML, Bootstrap et CSS. Les fonctionnalités attendues du site, ont été implémentés en PHP, afin de répondre à la problématique demandé par l’agence d’architecte.

Je vais, dans un premier temps, aborder la partie du site visible par les visiteurs, puis la partie administrateur du site.

V.1 - Partie visiteur

La partie visiteur qui est accessible à tous, est composée de pages web représentatives de l’agence, avec des représentations de travaux réalisés (éléments : images), des descriptions des activités (éléments : texte) et la possibilité de contacter l’agence (élément : formulaire).

Je vais prendre comme exemple la page Contact pour présenter le travail réalisé sur ce projet.

**Remarque :**

Chaque page est affectée du même entête, composé de photos et du logo de l’agence d’architecte et d’une barre de menu, qui permet d’accéder aux autres pages.

L’entête est codé dans deux fichiers :

- entete\_accueil.php : pour la mise en forme (annexe 1)

- entete\_accueil.css : pour la gestion du design (annexe 2).



**Figure 22 :** Entête du site

V.1.1 - Structuration de la page Contact

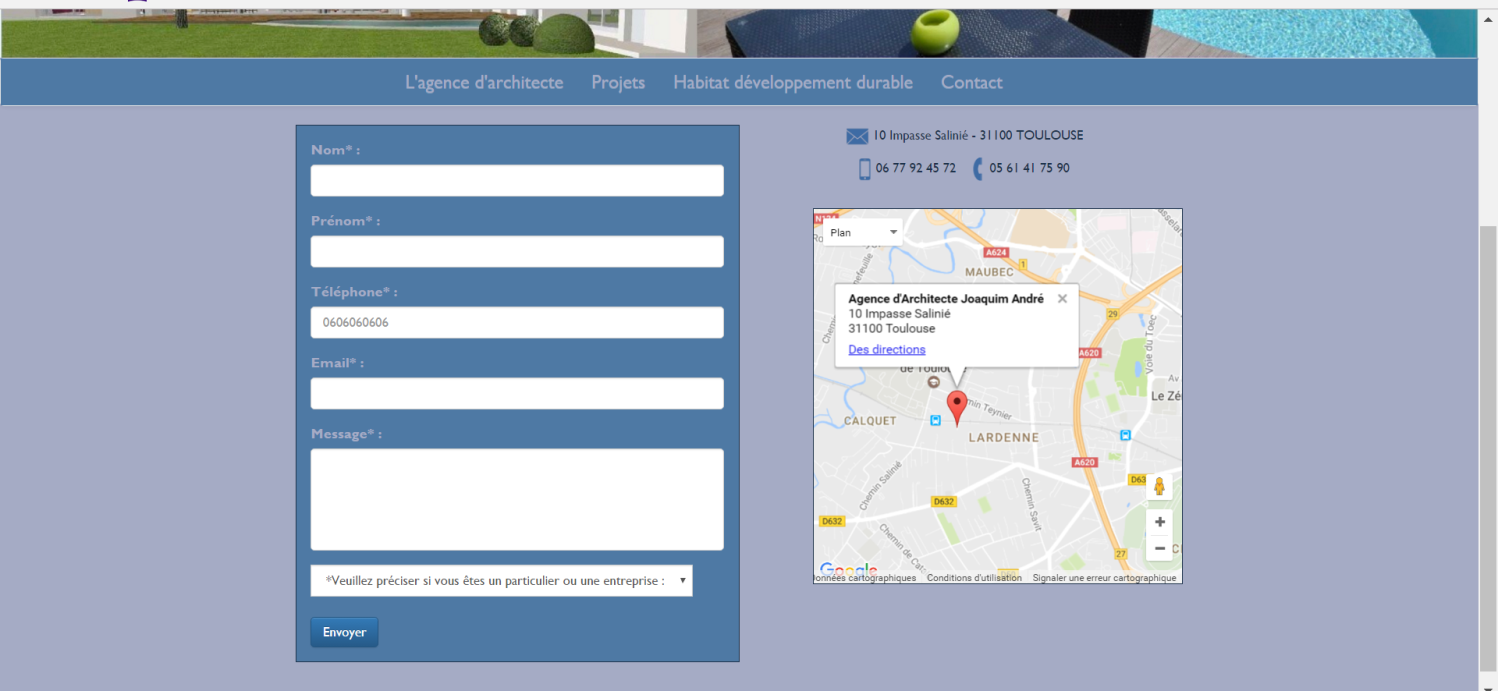
La structure de cette page doit répondre à des fonctionnalités tels que :

- un formulaire qui permet à un visiteur de prendre contact avec l’agence en renseignant les champs nom, prénom, téléphone, email, statut (particulier/entreprise) et en laissant un message

- un cadre texte + pictogrammes qui contient les coordonnées de l’agence

- un iframe qui comporte une Google Map.

Le HTML a permis de réaliser la structure, et la mise en page a été faite grâce à Bootstrap et au CSS. La figure ci-dessous montre le rendue de cette page.

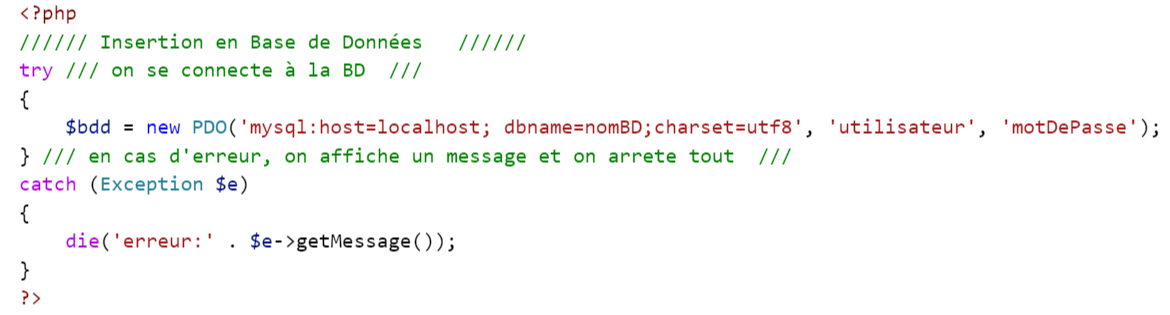


**Figure 22 :** Page Contact du site

V1.2 - Fonctionnalités de la page Contact

Le fonctionnement du formulaire a été réalisé en PHP, ce qui a permis de récupérer les données laissées par les visiteurs et de pouvoir les stocker dans la base de données.

Le stockage de ces données a demandé en amont une connexion à la base de données. Le code PHP suivant permet de réaliser cette connexion :



**Figure 23 :** Code PHP pour se connecter à la base de données

**&&&&&&&&&&& récuperation des données&&&&&&&&**

**Remarque :**

Pour inclure l’entête dans cette page j’ai utilisé une fonction PHP qui sollicite le fichier entete\_accueil.php :



**Figure 24 :** Code PHP pour inclure l’entête à la page