

République Algérienne Démocratique et Populaire  
Ministre de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Abderrahmane Mira - Béjaia -  
Faculté des Sciences Exactes  
Département d'informatique



*Mémoire de fin de cycle en vue de l'obtention d'un master professionnel  
option : Génie logiciel*

## Thème

Conception et réalisation d'une application web pour  
la gestion des annonces

Réalisé par :

M<sup>r</sup> Boumaza Lotfi

M<sup>r</sup> Zoutat Massinissa

Évalué par :

Examinatrice	M <sup>lle</sup> Chelouah Leila	M.C.B	U.A/Mira Béjaia, Algérie.
Examinatrice	M <sup>me</sup> Yessad Nawel	M.C.B	U.A/Mira Béjaia, Algérie.
Encadrante	M <sup>me</sup> Lahlah Souaad	M.C.B	U.A/Mira Béjaia, Algérie.

Promotion 2019-2020

## *- Remerciements -*

Nous tenons à remercier en premier lieu le bon Dieu de nous avoir donné la force et le courage afin de mener à terme ce travail.

Notre profonde gratitude et sincères remerciements vont à notre promotrice Mme LAHLAH Souaad pour son suivi, sa disponibilité, son orientation et ses conseils.

Nous remercions également, les membres du jury qui ont accepté d'examiner et de juger ce modeste travail.

Enfin, nos profonds remerciements vont à nos chers parents, frères et sœurs, pour leurs soutiens et leur confiance en nous, sans oublier nos amis et nos camarades en particulier, notre promotion de 2<sup>ème</sup> année MASTER (2019/2020). Et tous ceux qui ont contribué, de près ou de loin à la réalisation de notre projet.

*- Dédicaces -*

Je dédie cet humble travail à mes très chers parents, à toute ma famille, mes amis et à toutes les personnes qui m'ont aidé à le réaliser.

**Boumaza Lotfi**

Je dédie ce modeste travail à mes chers parents, frères, sœurs, ma bien aimée et à tous ceux qui ont contribué à sa réalisation.

**Zoutat Massinissa**

# Table des matières

<b>Table des matières</b>	<b>i</b>
<b>Liste des figures</b>	<b>iv</b>
<b>Liste des tableaux</b>	<b>vi</b>
<b>Liste des abréviations</b>	<b>vii</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Étude préalable</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Présentation du projet . . . . .	3
1.2.1 Objectifs . . . . .	3
1.2.2 Besoins fonctionnels . . . . .	4
1.2.3 Besoins non fonctionnels . . . . .	4
1.2.4 Vue globale de l'application à réaliser . . . . .	5
1.2.5 Le logo . . . . .	5
1.2.6 Analyse concurrentielle . . . . .	6
1.2.7 Discussion sur l'application par rapport à l'analyse concurrentielle . . . . .	7
1.3 Conclusion . . . . .	7
<b>2 Étude conceptuelle</b>	<b>8</b>
2.1 Introduction . . . . .	8
2.2 Le choix de la méthode . . . . .	8
2.2.1 Définition d'un processus . . . . .	8
2.2.1.1 Définition du processus unifié . . . . .	9
2.2.1.2 Les caractéristiques du processus unifié . . . . .	9
2.2.1.3 Cycle de vie du processus unifié . . . . .	10
2.2.1.4 Les activités . . . . .	11
2.2.1.5 Les phases . . . . .	13
2.3 Analyse conceptuelle . . . . .	14
2.3.1 Le langage UML . . . . .	14

2.3.1.1	Points forts d'UML . . . . .	15
2.3.1.2	Points faibles d'UML . . . . .	15
2.3.1.3	Les diagrammes . . . . .	15
2.3.2	Identification des acteurs . . . . .	18
2.3.3	Diagrammes des cas d'utilisations . . . . .	18
2.3.4	Modélisation des diagrammes de séquence . . . . .	21
2.3.5	Diagramme de classe . . . . .	28
2.3.6	Le schéma relationnel . . . . .	29
2.3.6.1	Règle de passage du diagramme de classe au modèle relationnel . . . . .	29
2.3.6.2	Le modèle relationnel . . . . .	30
2.3.7	Conclusion . . . . .	30
<b>3</b>	<b>Réalisation</b> . . . . .	<b>31</b>
3.1	Introduction . . . . .	31
3.2	Langages et environnements de développement . . . . .	31
3.2.1	Le modèle MVC . . . . .	31
3.2.2	PHP . . . . .	32
3.2.3	SQL . . . . .	32
3.2.4	HTML5 . . . . .	33
3.2.5	CSS3 . . . . .	33
3.2.6	Java Script . . . . .	33
3.2.7	jQuery . . . . .	33
3.2.8	Ajax . . . . .	33
3.3	Outils de développement . . . . .	34
3.3.1	XAMPP . . . . .	34
3.3.2	MYSQL . . . . .	34
3.3.3	PHPStorm . . . . .	34
3.3.4	Draw.io . . . . .	35
3.3.5	Photoshop . . . . .	35
3.4	Frameworks utilisés . . . . .	35
3.4.1	Bootstrap . . . . .	35
3.4.2	Laravel . . . . .	36
3.5	Présentation des interfaces graphiques . . . . .	41
3.5.1	La page d'accueil . . . . .	41
3.5.2	Page de pré-inscription . . . . .	42
3.5.3	Page d'authentification . . . . .	43
3.5.4	Les interfaces de publications d'annonces . . . . .	44
3.5.5	Les interfaces montrant l'affichage de l'annonce selon sa catégorie . . . . .	48
3.5.6	Page représentant les activités de l'administrateur . . . . .	51
3.6	Conclusion . . . . .	53

Conclusion générale	54
Bibliographie	55

# Table des figures

1.1	Vue globale de l'application à réaliser . . . . .	5
1.2	Logo de l'application . . . . .	5
1.3	Capture d'écran du site OuedKniss . . . . .	6
2.1	Déroulement du processus unifié [4] . . . . .	9
2.2	L'architecture bidirectionnelle [5] . . . . .	11
2.3	Illustration des différentes activités [6] . . . . .	13
2.4	Résumé du processus UP [6] . . . . .	14
2.5	Diagramme de cas d'utilisation « Internaute » . . . . .	19
2.6	Diagramme de cas d'utilisation « Annonceur » . . . . .	20
2.7	Diagramme de cas d'utilisation « Administrateur » . . . . .	21
2.8	Diagramme de séquence du cas d'utilisation « Se connecter » . . . . .	22
2.9	Diagramme de séquence du cas d'utilisation « pré-inscription » . . . . .	23
2.10	Diagramme de séquence du cas d'utilisation « Déposer une annonce » . . . . .	24
2.11	Diagramme de séquence du cas d'utilisation « Chercher une annonce » . . . . .	25
2.12	Diagramme de séquence du cas d'utilisation « Supprimer une annonce » . . . . .	26
2.13	Diagramme de séquence du cas d'utilisation « Modifier son compte » . . . . .	27
2.14	Diagramme de séquence du cas d'utilisation « Supprimer un utilisateur » . . . . .	28
2.15	Diagramme de classe . . . . .	29
3.1	Architecture MVC [9] . . . . .	32
3.2	Organisation de laravel . . . . .	37
3.3	Vue sur les Contrôleurs et Middleware . . . . .	38
3.4	Le dossier database . . . . .	39
3.5	Les différents models qui se trouvent dans app dont le model user . . . . .	40
3.6	Le dossier ressources . . . . .	40
3.7	Le dossier routes . . . . .	41
3.8	La page d'accueil . . . . .	42
3.9	La page de pré-inscription . . . . .	43
3.10	La page d'authentification . . . . .	44
3.11	Interface du choix de la catégorie . . . . .	45
3.12	Interface de publication d'annonce . . . . .	46
3.13	Interface de recherche d'annonce selon un produit . . . . .	47

3.14	Interface de recherche d'annonce selon la wilaya et la catégorie . . . . .	47
3.15	Les annonces de la catégorie promotions . . . . .	48
3.16	Les annonces de la catégorie vêtements . . . . .	49
3.17	Interface montrant les détails de l'annonce après avoir cliquer sur l'icône sous forme d'œil . . . . .	50
3.18	Les détails sur l'annonce . . . . .	50
3.19	L'interface d'accueil avant la suppression du produit « Samsung » . . . . .	51
3.20	Interface montrant la suppression de l'annonce après avoir cliquer sur la corbeille . . . . .	52
3.21	Interface montrant que l'annonce a été bien supprimée . . . . .	52



# Liste des tableaux

1.1	Résumé de l'analyse du site OuedKniss . . . . .	7
2.1	Identification des acteurs. . . . .	18
2.2	Les cas d'utilisation pour chaque acteur. . . . .	18

# Liste des abréviations

<b>UP</b>	<b>Unified Process</b>
<b>UML</b>	<b>Unified Modeling Language</b>
<b>MVC</b>	<b>Model View Controller</b>
<b>HTML</b>	<b>Hyper Text Markup Language</b>
<b>CSS</b>	<b>Cascading Style Sheets</b>
<b>JS</b>	<b>Java Script</b>
<b>AJAX</b>	<b>Asynchronous java script and xml</b>
<b>PHP</b>	<b>Personal Home Page</b>
<b>SQL</b>	<b>Structured Query Language</b>
<b>DOM</b>	<b>Document Object Model</b>
<b>OMG</b>	<b>Object Management Group</b>

# Introduction générale

De nos jours, la technologie se développe d'une manière remarquable dans tous les domaines et cela grâce à l'outil informatique. Ce dernier s'est glissé imperceptiblement dans la quasi-totalité de nos activités quotidiennes. Il joue un rôle important dans le développement de nombreuses entreprises et organisations via le développement des logiciels et des applications informatiques, donc l'informatique est omniprésente et surtout avec l'interaction de l'Internet qui est l'outil par excellence pour se faire connaître de tous.

Aujourd'hui internet est un outil d'information et de communication en pleine évolution offrant des perspectives de croissance exceptionnelles. C'est devenu un formidable moyen de communication, d'échange, de travail, de rencontre et même de **commerce** [1].

Grâce à l'Internet et à l'informatique, les entreprises (industrielles ou commerciales) ont pensé à une commercialisation plus rapide et plus moderne, d'où la conception de nouvelles méthodologies de commercialisation à savoir **l'apparition d'application web d'annonces** [2].

Toutefois, les entreprises, les associations, les organisations et les particuliers, les internautes d'une façon générale rencontrent aujourd'hui des difficultés majeures dans la publication ou la recherche des différentes annonces, à savoir :

- La difficulté à vendre ou à acheter un produit.
- Comment trouver les clients ou les vendeurs les plus proches.
- Comment trouver la clientèle ciblée.
- Le souci de vouloir toujours vendre ou acheter au meilleurs prix.
- La volonté de trouver les meilleurs produits sur le marché.
- La difficulté de se faire connaître sur un marché.
- L'impact écologique des affiches d'annonces sur les murs et les panneaux publicitaires.

L'objectif principal de ce projet est de pallier les différents problèmes cités ci-dessus en réalisant une application web pour la gestion des différents types d'annonces.

Voici **SoukDzair**, l'application que nous avons conçue et réalisée, qui facilitera les échanges entre vendeurs et acheteurs. C'est une solution simple, élégante, flexible et disponible 24h/24, qui

permettra aux futurs utilisateurs de chercher ou de déposer des annonces en ligne en un clin d'œil et qui rendra leurs annonces visibles sur tout le territoire national.

Notre mémoire est organisé en trois chapitres :

Dans le premier chapitre, nous avons étudié les besoins de l'application à réaliser sous forme d'une étude préalable.

Dans le deuxième chapitre, nous avons présenté la conception de notre application que nous avons modélisé avec le langage UML.

Dans le troisième chapitre, nous avons présenté notre application, en mentionnant les outils de programmation que nous avons utilisés, quelques interfaces graphiques y seront présentés.

Notre travail s'est achevé par une conclusion générale et quelques perspectives.

# Chapitre 1

## Étude préalable

### 1.1 Introduction

De nos jours les désires ne sont plus comme avant, aujourd'hui nous nous focalisons sur la réalisation des besoins efficacement en un temps minimal, c'est pour cela que nous optons pour les applications qui offrent des services en ligne sans se déplacer et sans perdre beaucoup de temps.

Dans le cadre de notre travail, nous nous focalisons sur la gestion des annonces en ligne notamment sur la publication d'une annonce, l'objectif principale est de faciliter la tâche à l'internaute concernant ses achats, ses ventes, ses locations, etc.

### 1.2 Présentation du projet

Le sujet principal de ce travail repose sur le développement d'une application web de gestion des annonces, notamment le dépôt des différents types et catégories d'annonces par l'internaute.

#### 1.2.1 Objectifs

- Faciliter la publication et la recherche d'annonces.
- Trouver son client ou son produit le plus proche possible.
- Cibler la clientèle.
- Gagner du temps.
- Vendre ou acheter au meilleurs prix.
- Trouver le meilleur produit sur le marché.
- Faciliter l'achat et la vente entre particulier.
- L'application doit aussi permettre à :
  - Tout visiteur (internaute) : de consulter le site, de remplir le formulaire de pré-inscription, de se connecter et voir des annonces.

- Tout annonceur : de publier une annonce, modifier son profil, supprimer ou voir les annonces.
- L'administrateur : de gérer les utilisateurs et les annonces.

### 1.2.2 Besoins fonctionnels

Le système de gestion des annonces dont nous voulons se doter, doit être opérationnel, évolutif, convivial et offrant les options de gestion d'annonces nécessaires aux utilisateurs.

Pour cela, l'application à réaliser doit satisfaire les exigences de la totalité des utilisateurs. Nous présentons dans ce qui suit tout les besoins fonctionnels classés en espace.

Notre application est divisée en cinq grands espaces :

- **Espace d'accueil** : contient l'interface de démarrage de notre application, n'importe qui peut consulter cette page.
- **Espace d'inscription** : c'est l'interface qui permet à un utilisateur de créer son compte avec ses informations personnelles, un mot de passe et un identifiant. Le formulaire contient des champs qui concernent les informations que chaque utilisateur doit remplir.
- **Espace de publication des annonces** : après l'authentification, l'utilisateur peut publier ses annonces en cliquant sur le bouton publier, la page demande à l'utilisateur de choisir la catégorie de son annonce par exemple : Voitures et Motos, Vêtements, Mobiles et Tablettes, immobilier et autres.
- **Espace de gestion du profil de l'utilisateur** : le lien vers cette page ne s'affiche qu'après l'authentification de l'utilisateur, où il peut voir la liste de ses annonces, ou les informations concernant son compte.
- **Espace d'administration** : c'est uniquement l'administrateur de l'application qui peut accéder à cet espace, où il peut consulter et gérer la liste des utilisateurs et la liste des annonces publiées sur l'application.

### 1.2.3 Besoins non fonctionnels

Disposer d'une application web fonctionnelle et attrayante contribue nettement à la réussite d'une grande majorité de la société d'accueil. Afin de se distinguer de la concurrence, l'application doit être esthétiquement attrayante et sa navigation doit être fluide et efficace [2].

- Le système doit avoir une interface conviviale permettant un dialogue simple entre l'utilisateur et l'application.
- Le système doit être fiable et sécurisé
- Le menu doit être bien organisé.
- Les informations figurant dans l'application doivent être claires et précises.
- L'application doit fonctionner sur différents systèmes d'exploitation.

### 1.2.4 Vue globale de l'application à réaliser

L'application à réaliser permettra de développer un tableau de bord pour les utilisateurs et les administrateurs, création de différentes catégories d'annonces, publication d'annonces dans différentes catégories et wilaya, publications d'images en plusieurs parties pour les annonces, filtrer les annonces en fonction du produit, filtrer les annonces en fonction de la combinaison de la wilaya (et sa commune) et de la catégorie. L'application permettra aussi à tout visiteur de consulter le site, à tout annonceur de déposer et voir sa (ses) propre(s) annonce(s) et à l'administrateur de gérer les annonces.

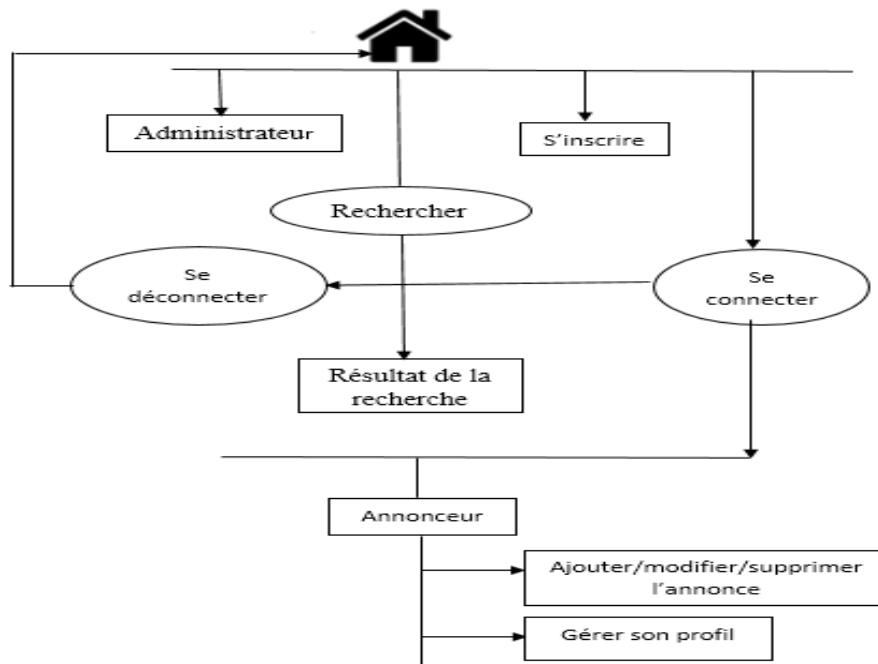


FIGURE 1.1 – Vue globale de l'application à réaliser

### 1.2.5 Le logo

Le logo choisi est une image représentant le nom de notre application web **SoukDzair**, un nom composé de «Souk» qui signifie le marché, et «Dzair» qui fait référence à l'Algérie, avec un panier blanc et arrière-plan rouge foncé, qui sont les couleurs majoritaires sur la plateforme.



FIGURE 1.2 – Logo de l'application

## 1.2.6 Analyse concurrentielle

Toutes plateformes, destinées aux annonces, offrent la possibilité aux utilisateurs de déposer des annonces commerciales, de vendre ou d'acheter pratiquement tout ce qu'ils souhaitent (conformément à la loi), dont chacune a ses propres avantages par rapport aux autres.

Prenant l'exemple du site algérien OuedKniss :

C'est le premier site web d'annonces et huitième le plus visités en Algérie. Créé en 2006 par un groupe d'étudiants algériens. Il permet de gérer les annonces des biens, produits et services, nous citons : automobiles, pièces détachées, emploi, divertissements, matériaux, équipement, média et immobilier. OuedKniss reçoit plus de 240 000 visiteurs par jour. Nous avons présenté dans la figure ci-dessous un aperçu du site.

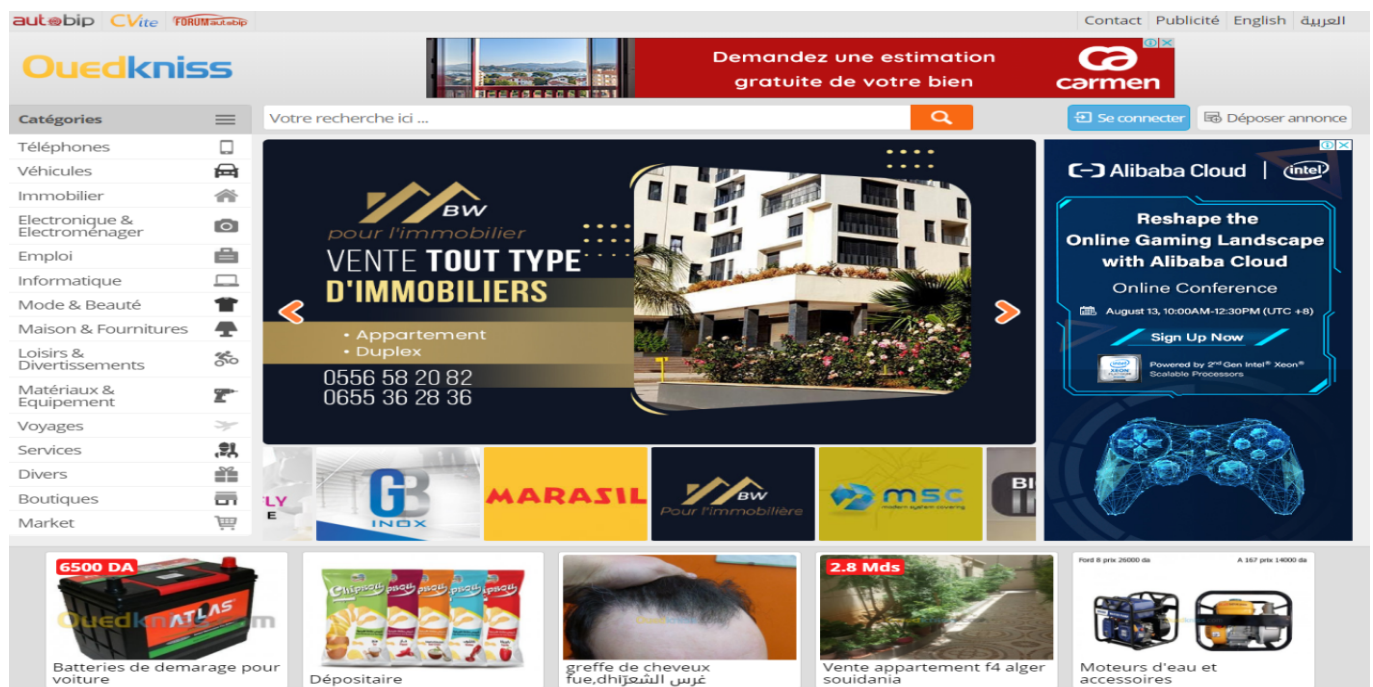


FIGURE 1.3 – Capture d'écran du site OuedKniss

Le tableau suivant résume les critères essentiels que nous avons pu cerner pendant notre analyse.



Nom du site	OuedKniss.com
Adresse du site	<a href="https://www.ouedkniss.com/">https ://www.ouedkniss.com/</a>
Date de l'analyse	10 avril 2020.
Technologie utilisé	PHP, JQuery.
Création	Août 2006.
Contenu	Petites annonces, Abonnements professionnels, Publicité
Points positifs	<ul style="list-style-type: none"> <li>• La plus grande plate-forme de petites annonces en Algérie.</li> <li>• Les pages web du site sont RESPONSIVE DESIGN c'est-à-dire qu'elles s'adaptent aux différents types d'équipements et périphériques.</li> <li>• Permet d'acheter et de vendre des produits et services dans des dizaines de catégories différentes.</li> <li>• S'engage à rester gratuit aux particuliers.</li> </ul>
Points négatifs	<ul style="list-style-type: none"> <li>• Ne contient pas d'annonces promotionnelles.</li> <li>• Présence remarquable de publicités.</li> <li>• Graphisme obsolète.</li> <li>• Le design semble bloqué dans les années 2000.</li> <li>• Annonces payantes pour les entreprises.</li> </ul>

TABLE 1.1 – Résumé de l'analyse du site OuedKniss

### 1.2.7 Discussion sur l'application par rapport à l'analyse concurrentielle

L'analyse que nous avons effectuée nous a permis de mieux définir les besoins et d'avoir une meilleure vue et représentation de notre application.

Le nom sur le site sera SoukDzair et son adresse internet sera [https ://www.SoukDzair.com](https://www.SoukDzair.com). Notre applications web s'adresse à un large public souhaitant publier une annonce, quel que soit le type d'annonce, tant que cette dernière est conforme à la loi.

## 1.3 Conclusion

À travers l'étude préalable, nous avons défini les besoins et les fonctionnalités qu'il va falloir implémenter à ces besoins, en effet la réalisation d'une application web nécessite une étude conceptuelle qui donnera une vision globale du futur système à implémenter et le chapitre suivant sera consacré à la conception.

# Chapitre 2

## Étude conceptuelle

### 2.1 Introduction

La phase de conception est la première étape dans la réalisation d'un projet, elle doit d'écrire de manière non ambiguë le fonctionnement futur du système, afin d'en faciliter la réalisation. Pour cela, différentes méthodes existent permettant de formaliser les étapes préliminaires du développement.

Ainsi dans ce chapitre, nous allons déterminer un premier découpage de notre application, et établir les différentes étapes de conception, nous définirons les relations qui peuvent exister entre les cas d'utilisation, et nous entamerons l'élaboration de certains diagrammes définis dans le langage **UML**(Unified Modeling Language), comme le diagramme de cas d'utilisation qui nous aidera dans la spécification des besoins de notre système, le diagramme de séquence qui nous donnera le scénario des actions et interactions entre les acteurs et notre système, et le diagramme de classe qui définira l'architecture statique de notre système.

### 2.2 Le choix de la méthode

Il existe plusieurs méthodes de développement logiciel construites sur UML comme : UP, RUP, TTUP, UPagile, XP, 2TUP, etc.

Parmi ses méthodes notre choix est basé sur la méthode UP (Unified Process).

#### 2.2.1 Définition d'un processus

Un processus définit une séquence d'étapes, en partie ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant.

L'objet d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles [3].

### 2.2.1.1 Définition du processus unifié

Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différents tailles de l'entreprise.

L'objectif d'un processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques [4].

UP répond aux préoccupations suivantes :

- QUI participe au projet ?
- Quoi, qu'est-ce qui est produit durant le projet ?
- COMMENT doit-il être réalisé ?
- QUAND est réalisé chaque livrable ?

### 2.2.1.2 Les caractéristiques du processus unifié

- *UP est itératif et incrémental*

Le projet est découpé en itérations ou étapes de courte durée qui permettent de mieux suivre l'avancement globale. À la fin de chaque itération une partie exécutable du système finale est produite, de façon incrémentale (par ajout).

La figure ci-dessous illustre l'interaction d'UP :

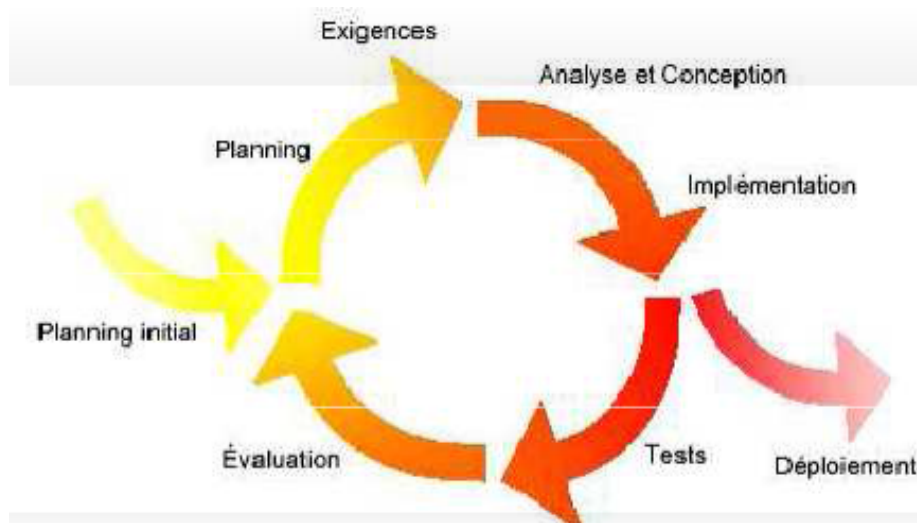


FIGURE 2.1 – Déroulement du processus unifié [4]

- *UP est centré sur l'architecture*

Tout système complexe doit être décomposé en partie modulaire afin d'en faciliter la maintenance et l'évolution. Cette architecture (fonctionnelle, logique, matérielle, etc) doit être modéliser en UML, et pas seulement documentée en texte.

- *UP est guidé par les cas d'utilisation d'UML*

Le but principal d'un système informatique est de satisfaire les besoins du client. Le processus de développement sera donc accès sur l'utilisateur. Les cas d'utilisation permettent d'illustrer ces besoins. Ils détectent puis décrivent les besoins fonctionnels et leur ensemble constitue le modèle de cas d'utilisation qui dicte les fonctionnalités complètes du système [4].

- *UP est piloté par les risques*

Les risques majeurs du projet doivent être identifiés au plus tôt mais surtout levés le plus rapidement. Les mesures à prendre dans ce cadre déterminant l'ordre des interactions [4].

### 2.2.1.3 Cycle de vie du processus unifié

L'objectif d'un processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques. UP est un ensemble de principes génériques adapté en fonctions des spécificités des projets :

- **L'architecture bidirectionnelle** : UP gère le processus de développement par deux axes (voir la figure 2.2).
- **L'axe vertical** : représente les principaux enchaînements d'activités, qui regroupent les activités selon leur nature. Cette dimension rend compte l'aspect statique du processus qui s'exprime en termes de composants, de processus, d'activités, d'enchaînements, d'artefacts et de travailleurs.
- **L'axe horizontal** : représente le temps et montre le déroulement du cycle de vie du processus ; cette dimension rend compte de l'aspect dynamique du processus qui s'exprime en terme de cycles, de phases, d'interactions et de jalons.

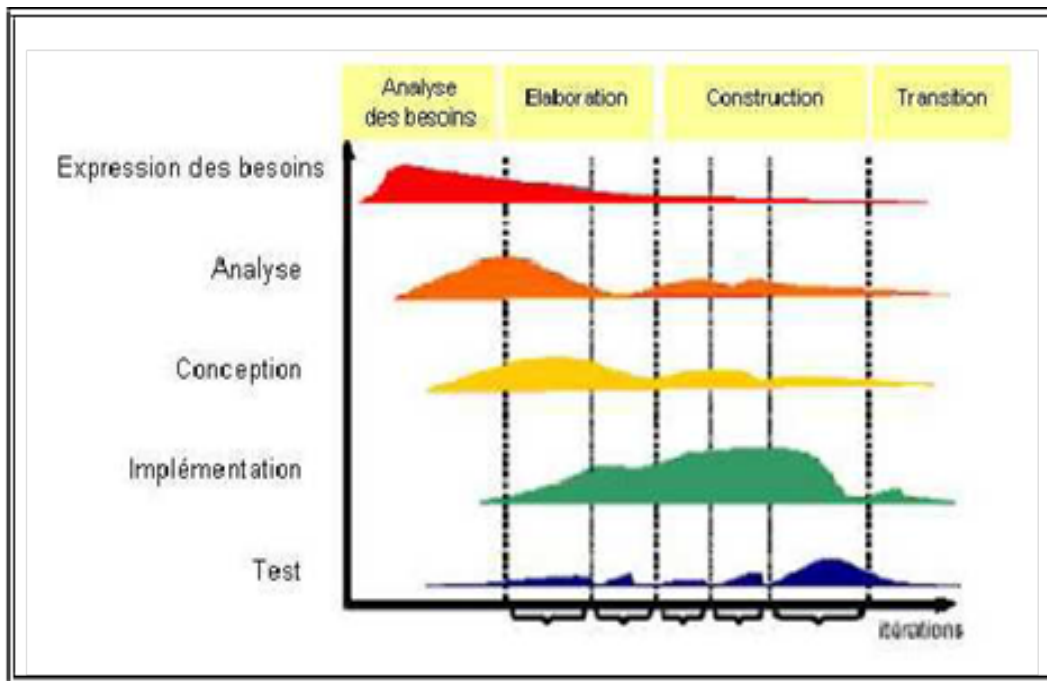


FIGURE 2.2 – L'architecture bidirectionnelle [5]

Pour mener efficacement un tel cycle, les développeurs ont besoins de toutes les représentations du produit logiciel, c'est-à-dire :

- Un modèle de cas d'utilisation.
- Un modèle d'analyse : détaillant les cas d'utilisation.
- Un modèle de conception : finissant la structure statique du système sous forme de sous-systèmes, de classes et d'interfaces.
- Un modèle d'implémentation : intégrant les composants.
- Un modèle de déploiement : définissant les nœuds physiques des ordinateurs.
- Un modèle de test : décrivant les cas de test vérifiant les cas d'utilisation.
- Une représentation de l'architecture [5].

#### 2.2.1.4 Les activités

- *Expression des besoins*

L'expression des besoins comme son nom l'indique, permet de définir les différents besoins :

- Inventorier les besoins principaux et fournir une liste de leurs fonctions.
- Appréhender les besoins non fonctionnels (techniques) et livrer une liste des exigences.
- Le modèle de cas d'utilisation présente le système du point de vue de l'utilisateur et représente sous forme de cas d'utilisation et d'acteur, les besoins du client.

- *Analyse*

L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences du client. Il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution.

Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structure sous une forme qui facilite la compréhension (scénarios), la préparation (définition de l'architecture), la modification et la maintenance du futur système. Il s'écrit dans le langage des développeurs et peut être considéré comme une première ébauche du modèle de conception.

- *Conception*

La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et au système d'exploitation. Elle détermine les principales interfaces et les transcrit à l'aide d'une notation commune.

Elle constitue un point de départ à l'implémentation :

- elle décompose le travail d'implémentation en sous-système.
- elle crée une abstraction transparente de l'implémentation.

- *Implémentation*

L'implémentation est le résultat de la conception pour implémenter le système sous formes de composants, c'est-à-dire, de code source, de scripts, de binaires, d'exécutable et d'autres éléments du même type.

Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération, et de produire les classes et les sous-systèmes sous formes de codes sources.

- *Test*

Les tests permettent de vérifier des résultats de l'implémentation en testant la construction. Pour mener à bien ces tests, il faut les planifier pour chaque itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun [6].

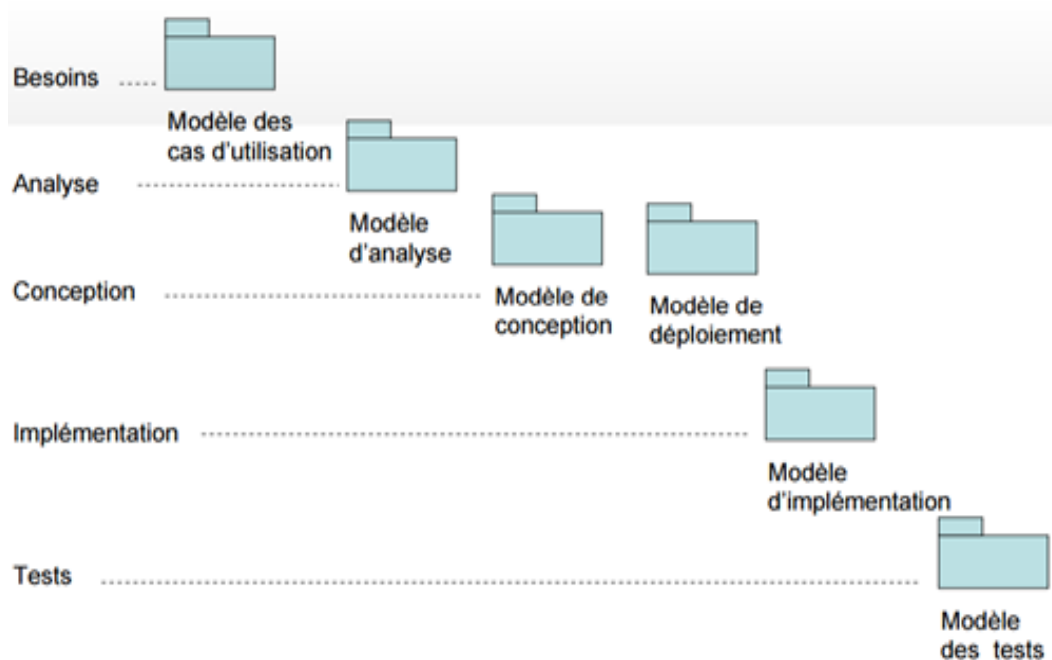


FIGURE 2.3 – Illustration des différentes activités [6]

### 2.2.1.5 Les phases

- *Analyse des besoins*

L'analyse des besoins donne une vue du projet sous forme de produit fini. Cette phase porte essentiellement sur les besoins principaux (du point de vue de l'utilisateur), l'architecture générale du système, les risques majeurs, les délais et les coûts.

- *Élaboration*

L'élaboration reprend les éléments de la phase d'analyse des besoins et les précise pour arriver à une spécification détaillée de la solution à mettre en œuvre.

L'élaboration permet de préciser la plupart des cas d'utilisation, de concevoir l'architecture du système et surtout de déterminer l'architecture de référence.

- *Construction*

La construction est le moment où l'on construit le produit. L'architecture de référence se métamorphose en produit complet. Le produit contient tous les cas d'utilisation que les chefs de projet, en accord avec les utilisateurs ont décidé de mettre au point pour cette version.

- *Transition*

Le produit est en version bêta. Un groupe d'utilisateurs essaye le produit et détecte les anomalies et défauts. Cette phase suppose des activités comme la formation des utilisateurs clients, la mise en œuvre d'un service d'assistance et la correction des anomalies constatées.

Tout simplement la phase de transition permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux [6].

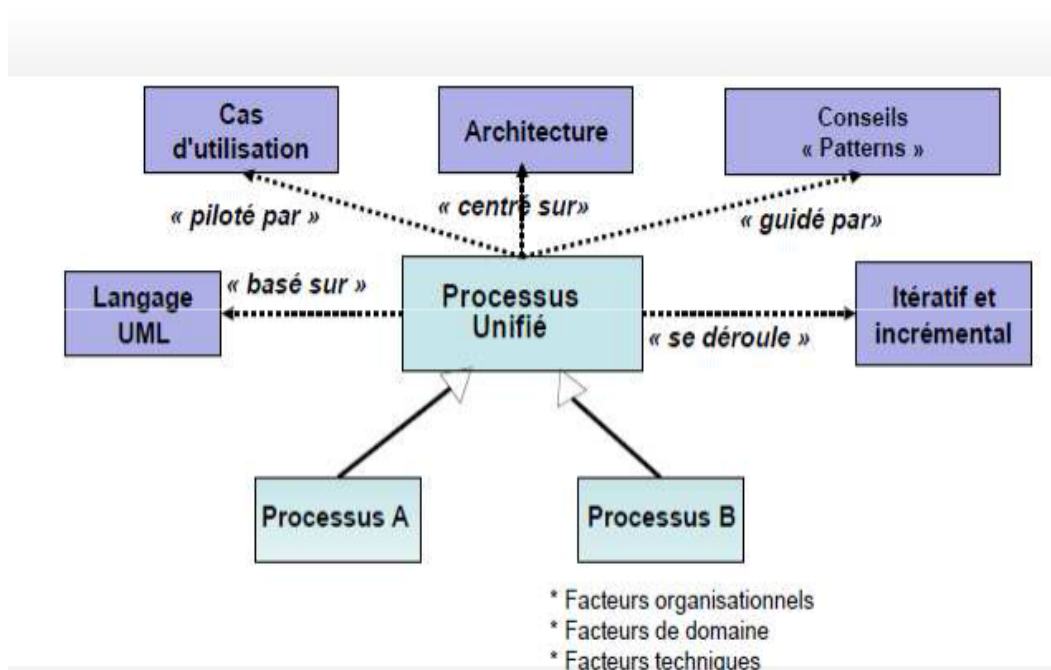


FIGURE 2.4 – Résumé du processus UP [6]

## 2.3 Analyse conceptuelle

UML se définit comme un langage de graphique destiné à la représentation d'un système donné selon un point de vue particulier.

### 2.3.1 Le langage UML

C'est un langage de modélisation graphique à base de pictogrammes, conçu pour représenter et spécifier les artefacts de systèmes logiciels, de plus il est destiné à comprendre et décrire des besoins spécifiés et documentés des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue, comme il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

UML résulte de l'unification de techniques ayant fait leurs preuves pour l'analyse et la conception de grands logiciels et de systèmes complexes.

- *UML est une norme*

Il est nécessaire qu'une méthode objet soit définie de manière rigoureuse et unique afin de lever les ambiguïtés. De nombreuses méthodes objet ont été définies, mais aucune n'a su s'imposer en raison du manque de standardisation. C'est pourquoi l'ensemble des acteurs du monde informatique



a fondé en 1989 l'OMG (Object Management Group), une organisation à but non lucratif, dont le but est de mettre au point des standards garantissant la compatibilité entre des applications programmées à l'aide de langages objet et fonctionnant sur des réseaux hétérogènes (de différents types). Et depuis 1997, UML est devenue une norme de l'OMG, ce qui lui a permis de s'imposer autant que méthode de développement objet et être reconnue et utilisée par de nombreuses entreprises.

- *UML est un langage de modélisation objet*

UML comble une lacune importante des technologies objet, il permet d'exprimer, d'élaborer et de modéliser au sens de la théorie des langages, de ce fait il contient les éléments constitutifs de ces derniers : concepts, une syntaxe et une sémantique.

- *UML décrit un méta modèle*

La puissance et l'intérêt d'UML est qu'il normalise la sémantique des concepts qu'il véhicule, il repose sur un méta modèle pour permettre à n'importe qui de déchiffrer son intention de manière non équivoque, il est donc primordial de s'accorder sur la sémantique des éléments de modélisation, bien avant de s'intéresser à la manière de les présenter [7].

### 2.3.1.1 Points forts d'UML

Il permet ainsi :

- Un gain de précision.
- Un gage de stabilité.
- L'utilisation d'outils.
- Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel [7].

### 2.3.1.2 Points faibles d'UML

- La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.
- L'intégration d'UML dans un processus n'est pas triviale, et améliorer un processus est une tâche complexe et longue [7].

### 2.3.1.3 Les diagrammes

#### A) Définition d'un diagramme

Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle. Chaque type de diagramme UML possède une structure et véhicule une sémantique précise.

## B) Les différents types de diagrammes

Pour notre projet, nous présenterons trois diagrammes d'UML : cas d'utilisation, de séquence et de classe.

- *Diagramme de classe (class diagram)*

Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre ses classes. Outre les classes, ils représentent un ensemble d'interfaces et de paquetages, ainsi que leurs relations.

Les diagrammes de classes contiennent généralement les éléments suivant :

- Les classes** : Une classe est la description d'un ensemble d'objet qui partage les mêmes attributs, les mêmes opérations, les mêmes relations et la même sémantique. Une classe est symbolisée par un rectangle.
- Attribut** : Un attribut est une propriété nommée d'une classe qui décrit un ensemble de valeurs que les instances de cette propriété peuvent prendre. Une classe peut ne pas avoir, comme elle peut avoir un ou plusieurs attributs.
- Opérations** : Une opération est une abstraction de ce que peut réaliser un objet et qui est réalisable par tous les objets de la classe. Une classe peut ne pas avoir comme elle peut avoir plusieurs opérations.
- Les relations d'association, d'agrégation et de composition** :
  - Une association représente une relation sémantique durable entre deux classes.
  - Une agrégation est un cas particulier d'association non symétrique exprimant une relation de contenance.
  - Une composition est une agrégation plus forte [8].

- *Diagramme de cas d'utilisation (use case diagram)*

Les diagrammes de cas d'utilisation représentent un ensemble de cas d'utilisation, d'acteurs et leurs relations. Ils représentent la vue statique des cas d'utilisation d'un système et sont particulièrement importants dans l'organisation et la modélisation des comportements d'un système.

- Les cas d'utilisation** : Les cas d'utilisation décrivent, sous la forme d'actions et de réactions, le comportement, ou tout simplement ce que fait le système du point de vue de l'utilisateur, encore appelé acteur. On recense, de la sorte, l'ensemble des fonctionnalités d'un système en examinant les besoins fonctionnels de chaque acteur.
- Les acteurs** : Un acteur représente un ensemble cohérent de rôles joués par les utilisateurs des cas d'utilisation en interaction avec ces cas d'utilisation. En règle générale, un acteur représente un rôle qu'un homme, une machine ou même un autre système joue avec le système. Il existe 4 grandes catégories d'acteurs :

- Les acteurs principaux : cette catégorie regroupe les personnes qui utilisent les fonctions principales du système.
- Les acteurs secondaires : cette catégorie regroupe les personnes qui effectuent des tâches administratives ou de maintenance.
- Le matériel externe : cette catégorie regroupe les dispositifs matériels autres que les ordinateurs comme les périphériques.
- Les autres systèmes : cette catégorie regroupe les systèmes avec lesquels le système interagit.
- **Les relations entre les cas d'utilisation** : UML définit trois types de relations standardisées entre cas d'utilisation, détaillées ci-après :
  - **La relation d'inclusion** : formalisée par le mot clé « include », le cas d'utilisation de base en incorpore explicitement un autre de façon obligatoire.
  - **La relation d'extension** : formalisée par le mot clé « extend », le cas d'utilisation de base en incorpore explicitement un autre, de façon optionnelle.
  - **La relation de généralisation ou spécialisation** : les cas d'utilisation descendants héritent de la description de leur parent commun. Chacun d'entre eux peut néanmoins comprendre des interactions spécifiques supplémentaires [8].
- *Diagramme de séquence (sequence diagram)*

Un diagramme de séquence met en évidence le classement des messages par ordre chronologique. On forme un diagramme de séquence en plaçant d'abord les objets qui participent à l'interaction en haut du diagramme sur l'axe des abscisses. En générale, on place l'objet qui débute l'interaction à gauche, puis on continue en progressant vers la droite, les objets les plus subordonnés étant tout à fait à droite. On place ensuite les messages envoyés et reçus par ces objets le long de l'axe des ordonnées, par ordre chronologique, du haut vers le bas. Cela donne au lecteur une indication visuelle claire du flot de contrôle dans le temps.

En générale, les diagrammes de séquence contiennent :

- **L'objet** : est une manifestation concrète d'une abstraction à laquelle on peut appliquer un ensemble d'opérations et qui possède un état capable de mémoriser les effets de ces opérations. On représente un objet en soulignant son nom.
- **Le lien** : est une liaison sémantique entre objets, en générale, il s'agit d'une instance d'une association. Chaque fois qu'une classe est reliée à une autre par une association, il peut y avoir un lien entre les instances des deux classes, et chaque fois qu'un lien existe entre deux objets, le premier objet peut envoyer un message au deuxième.
- **Le message** : est la spécification d'une communication entre objets, qui transporte des informations et qui s'affiche dans le but de déclencher une activité. La réception d'une instance de message peut être considérée comme une instance d'un événement [8].

### 2.3.2 Identification des acteurs

Après avoir recensé l'ensemble des besoins (cahier des charges), nous avons distingué les différents acteurs suivants :

Acteur	Codification	Rôle
Administrateur	Admin	Gérer les utilisateurs, ajouter ou supprimer des annonces.
visiteur anonyme	Internaute	Consulter le site, remplir le formulaire de pré-inscription, se connecter et voir des annonces.
Les inscrits	Annonceurs	Publier une annonce, consulter une annonce, rechercher une annonce, modifier son compte, supprimer ou modifier ses annonces.

TABLE 2.1 – Identification des acteurs.

### 2.3.3 Diagrammes des cas d'utilisations

Les cas d'utilisation sont des représentations fonctionnelles du système. Ils permettent de modéliser les attentes des utilisateurs dans le but d'effectuer une bonne délimitation du système et également la compréhension de son fonctionnement.

À partir des besoins exprimés, nous avons déduit les cas d'utilisation tel que résumés dans le tableau ci-dessous :

n°	Cas d'utilisation	Acteur
1	Pré-inscription	Internaute
2	Se connecter	
3	Rechercher une annonce	Internaute et annonceur
4	Consulter des annonce	
5	Publier une annonce	Annonceur
6	Modifier son annonce	
7	Supprimer son annonce	
8	Modifier son compte	
9	Ajouter annonces ou utilisateurs	Administrateur
10	Supprimer annonces ou utilisateurs	

TABLE 2.2 – Les cas d'utilisation pour chaque acteur.

- Spécification des cas d'utilisation

Diagramme de cas d'utilisation « Internaute »

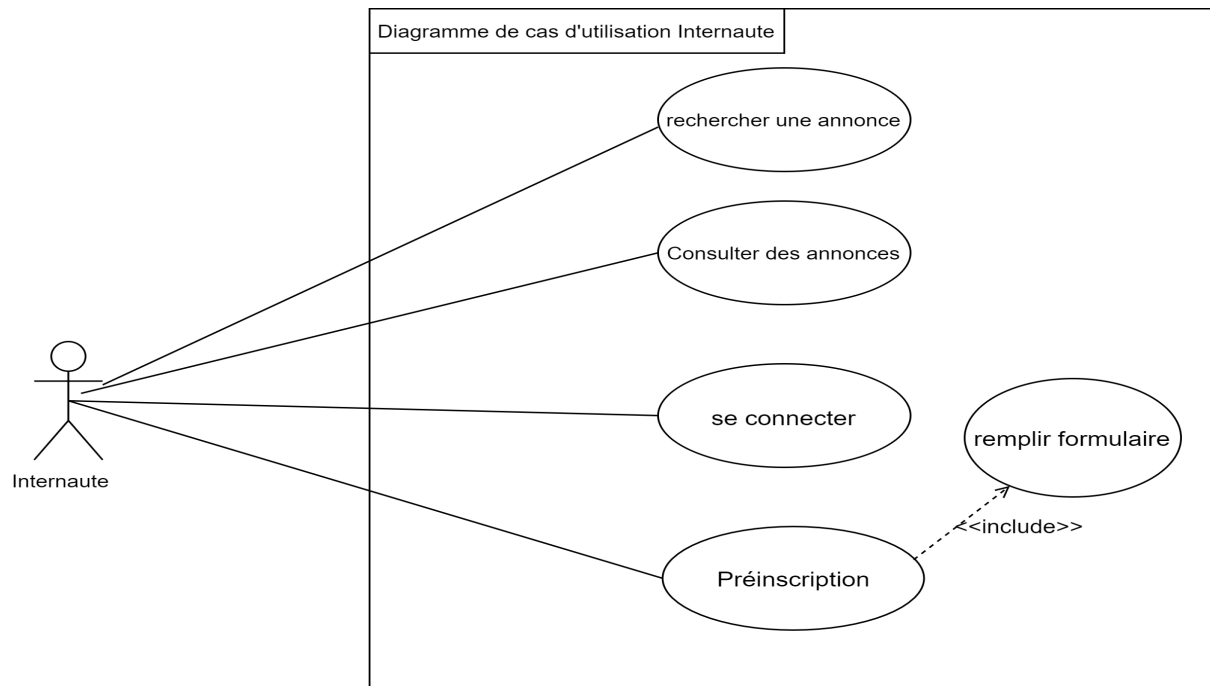


FIGURE 2.5 – Diagramme de cas d'utilisation « Internaute »

Diagramme de cas d'utilisation « Annonceur »

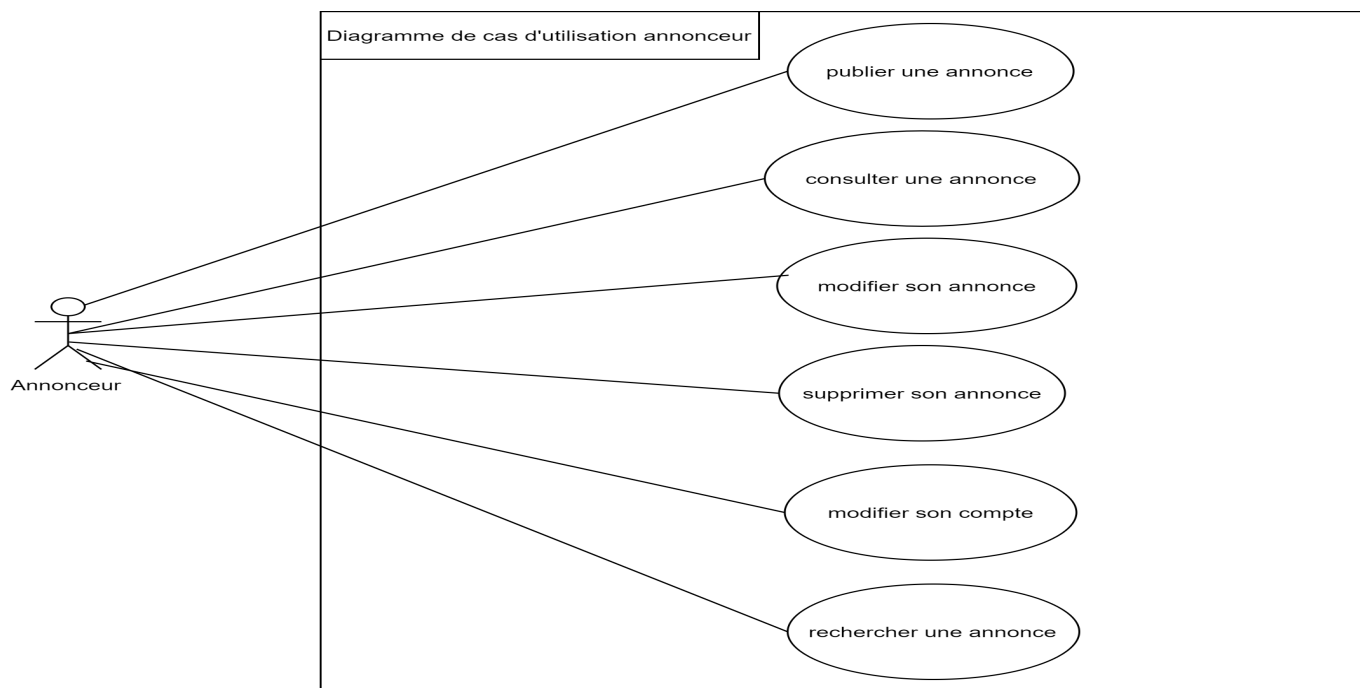


FIGURE 2.6 – Diagramme de cas d'utilisation « Annonceur »

## Diagramme de cas d'utilisation « Administrateur »

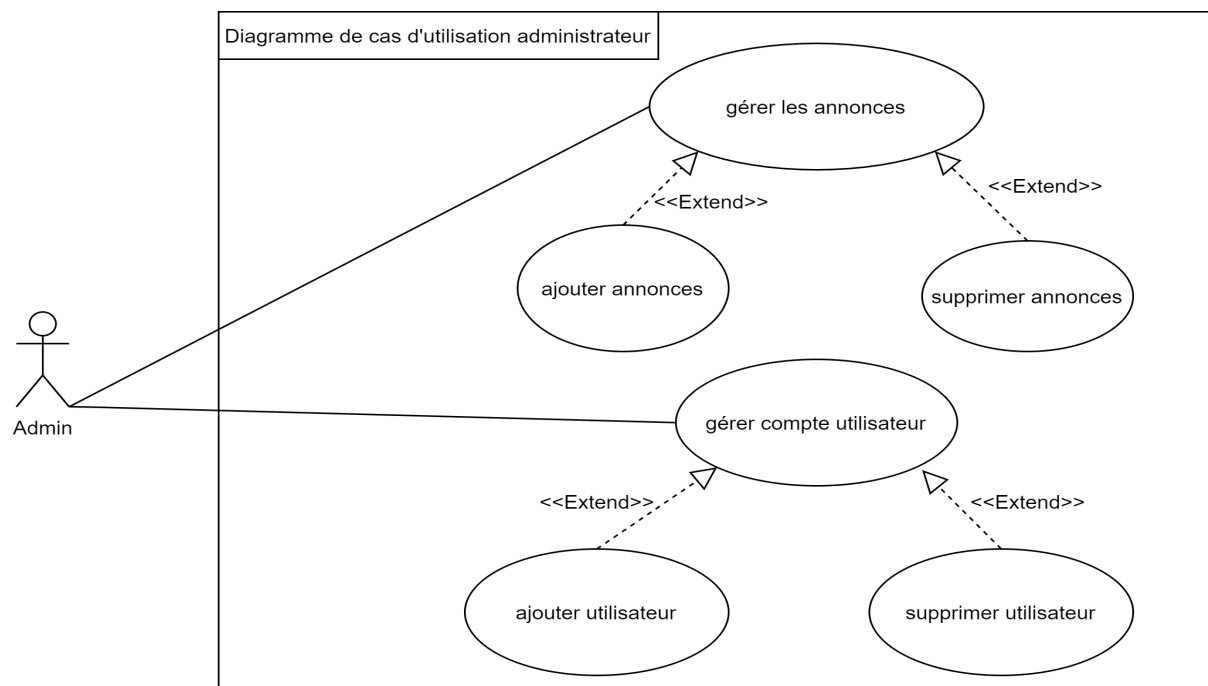


FIGURE 2.7 – Diagramme de cas d'utilisation « Administrateur »

## 2.3.4 Modélisation des diagrammes de séquence

Un diagramme de séquences est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées, avec une représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

- Se connecter

Le diagramme de séquence suivant illustre les interactions nécessaires pour la connexion d'un internaute. Après l'inscription utilisateur peut se connecter à l'application, pour cela il n'a qu'à cliquer sur le bouton de connexion en saisissant son identifiant et son mot de passe, le système vérifie ses champs auprès de la base de données, si ses informations sont correctes alors il aura accès à l'application pour gérer les annonces comme par exemple ajouter une annonce sinon un message d'erreur sera affiché et le système lui redemande à nouveau de saisir ses coordonnées.

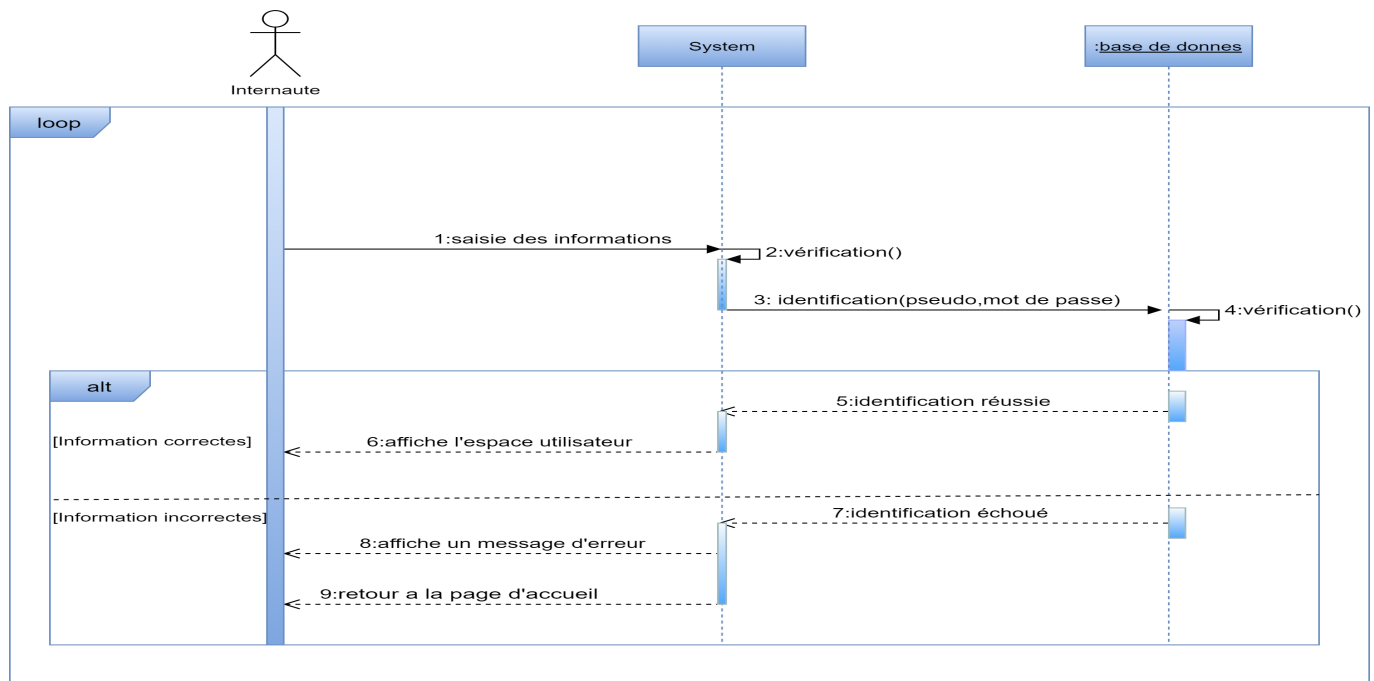


FIGURE 2.8 – Diagramme de séquence du cas d’utilisation « Se connecter »

- Pré-inscription

Le diagramme de séquence suivant illustre les interactions nécessaires pour l’inscription d’un utilisateur. En effet l’internaute va demander l’inscription auprès du système, ce dernier lui affiche un formulaire de pré-inscription, l’internaute remplit le formulaire de pré-inscription et le valide.

Les informations saisies par l’utilisateur sont traitées, c’est-à-dire le système va vérifier si le formulaire est complet et si les données sont cohérentes, dans le cas contraire un message d’erreur sera affiché.



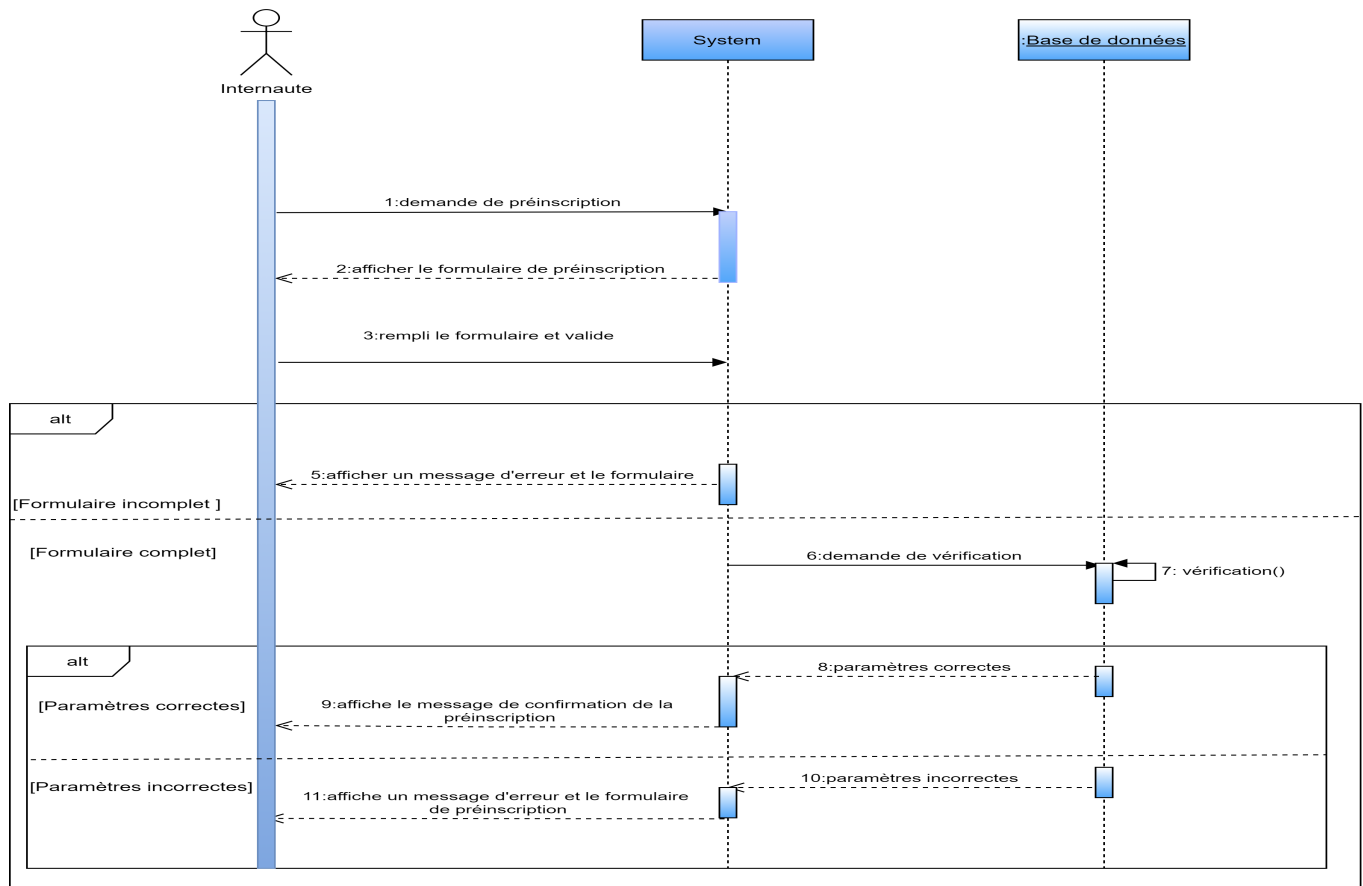


FIGURE 2.9 – Diagramme de séquence du cas d'utilisation « pré-inscription »

- Publier une annonce

Le diagramme de séquence suivant illustre les interactions nécessaires pour la publication d'une annonce. Après l'authentification, l'annonceur peut déposer une annonce tout en remplissant un formulaire de dépôt d'une annonce. Si l'annonceur oublie de remplir l'un des champs ou remplit un champs dans une case inapproprié l'annonce ne sera pas déposée.

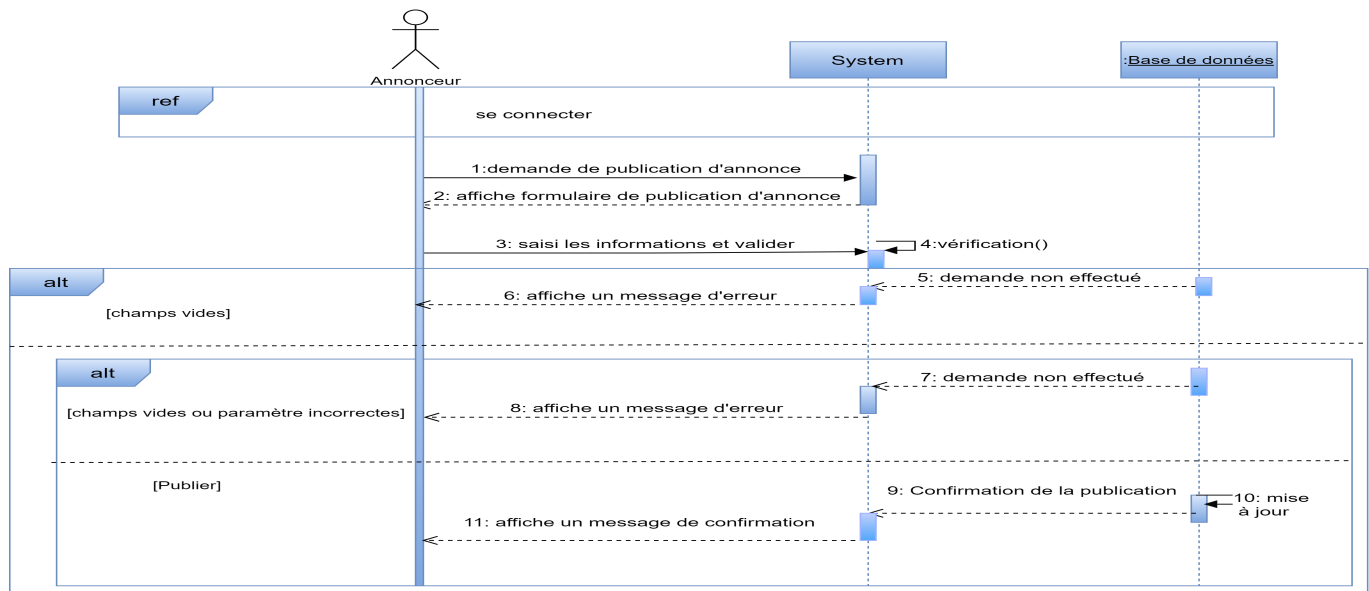


FIGURE 2.10 – Diagramme de séquence du cas d'utilisation « Déposer une annonce »

- Chercher une annonce

Le diagramme de séquence suivant illustre les interactions nécessaires pour la recherche d'une annonce. En effet, l'internaute saisit la recherche, le système va vérifier si le produit recherché existe dans la base de données ou pas, et il lui affiche un résultat.

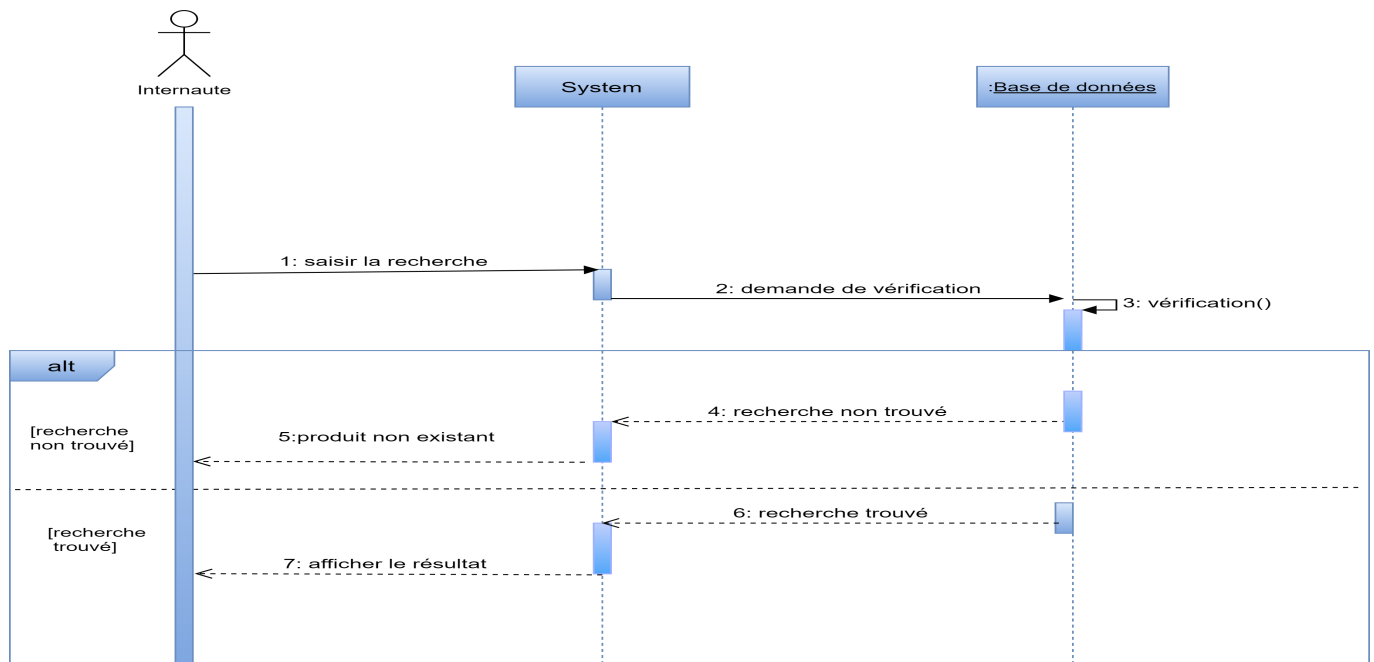


FIGURE 2.11 – Diagramme de séquence du cas d’utilisation « Chercher une annonce »

- Supprimer une annonce

Le diagramme de séquence suivant illustre les interactions nécessaires pour la suppression d’une annonce. En effet l’annonceur choisit une annonce, le système demande à l’annonceur une confirmation de suppression, si oui, le système vérifie l’existence de l’annonce dans la base de données, si l’annonce existe la suppression va être appliquée sinon un message d’erreur va être affiché.

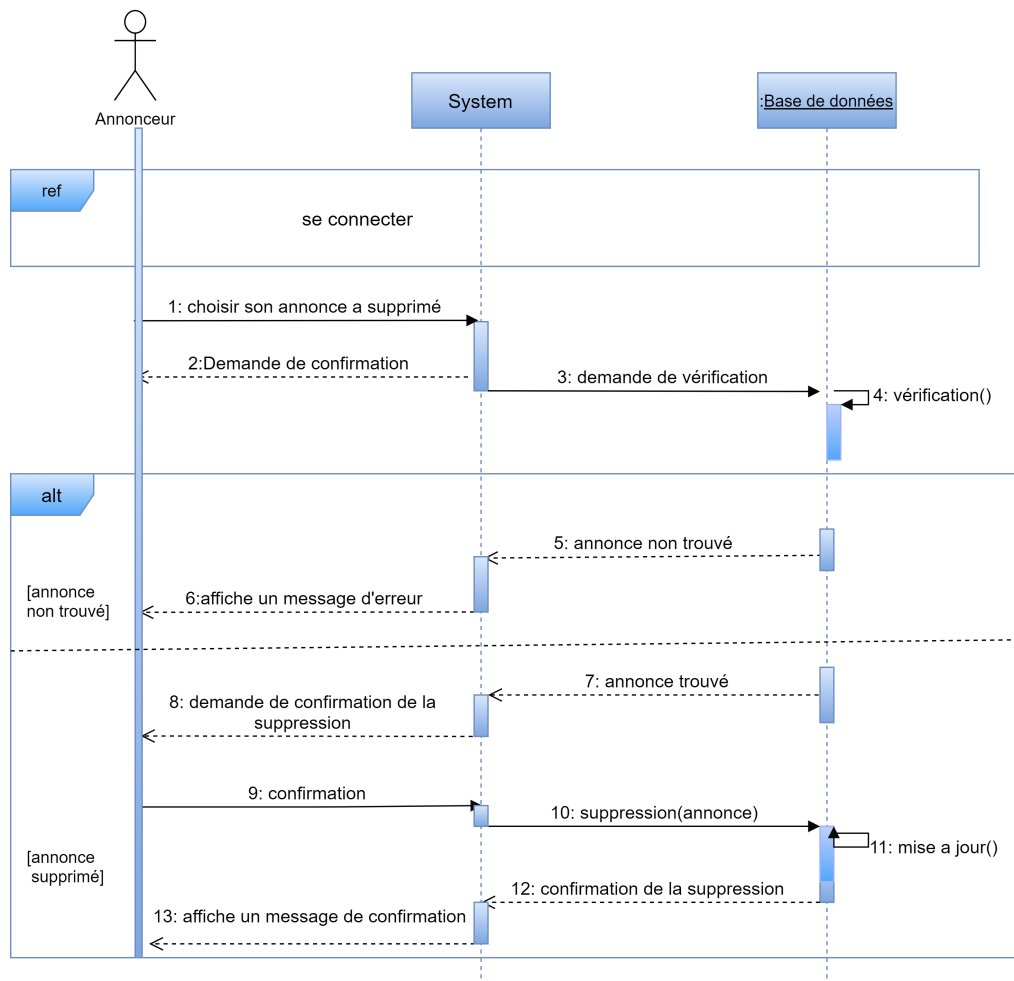


FIGURE 2.12 – Diagramme de séquence du cas d'utilisation « Supprimer une annonce »

- Modifier son compte

Le diagramme de séquence suivant illustre les interactions nécessaires pour qu'un annonceur modifie les informations de son compte. En effet, l'annonceur demande au système la modification de son compte, ensuite le système affiche le formulaire de modification et l'annonceur modifie ses informations, le système vérifie à nouveau les informations, si elles sont correctes, la base de données sera mise à jour sinon un message d'erreur sera affiché.

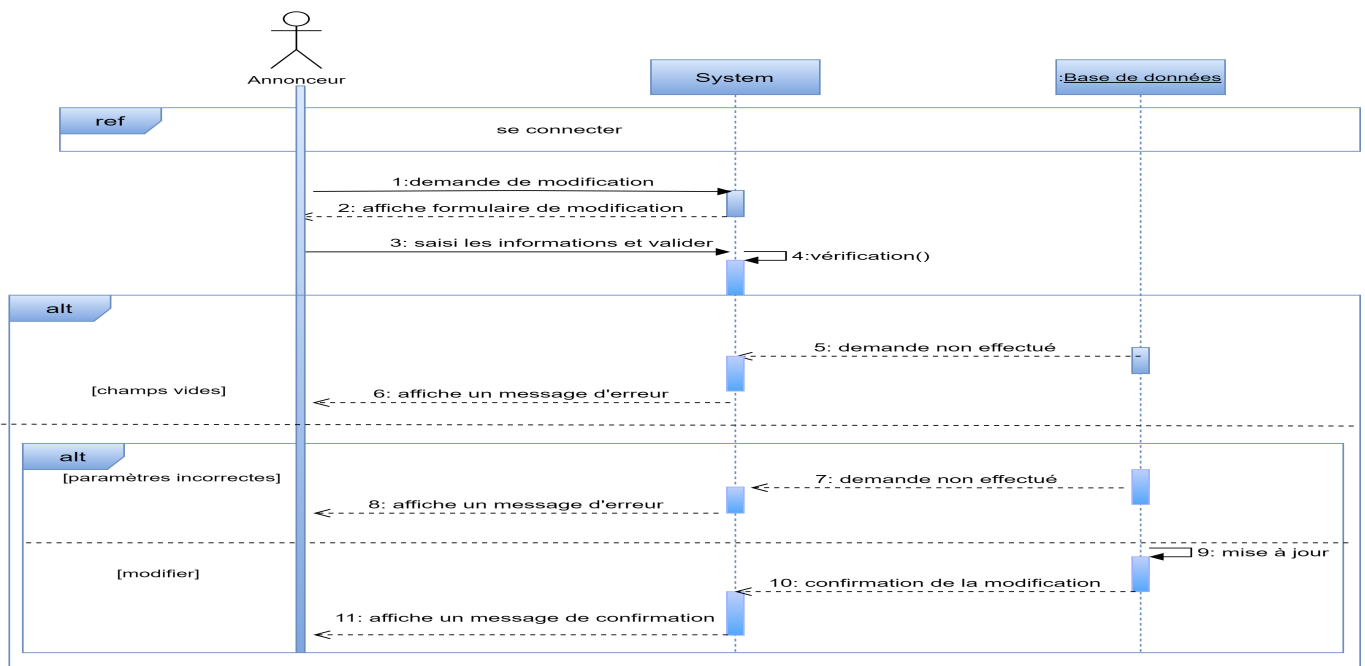


FIGURE 2.13 – Diagramme de séquence du cas d'utilisation « Modifier son compte »

- Supprimer un utilisateur

Le diagramme de séquence suivant illustre les interactions nécessaires pour la suppression d'un utilisateur. En effet, l'administrateur choisit l'utilisateur à supprimer, le système vérifie son existence dans la base de données, si l'utilisateur existe la suppression sera appliquée sinon un message d'erreur sera affiché.

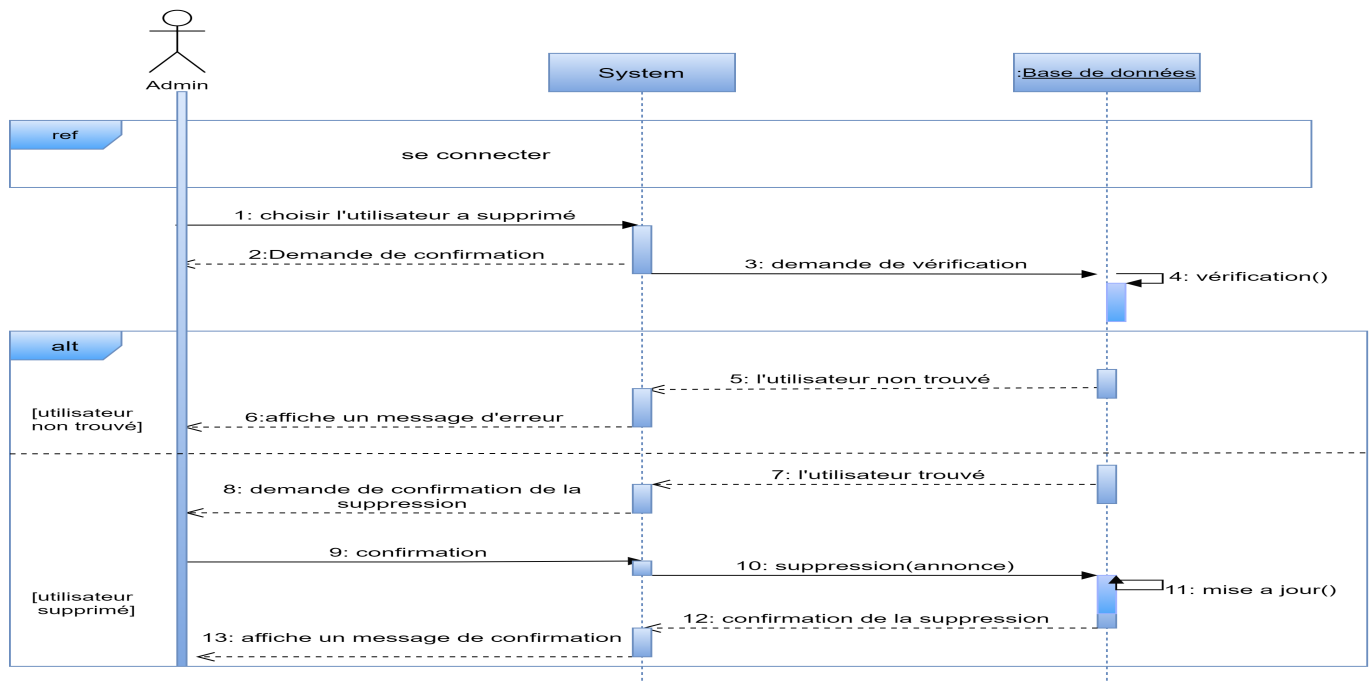


FIGURE 2.14 – Diagramme de séquence du cas d'utilisation « Supprimer un utilisateur »

### 2.3.5 Diagramme de classe

Le diagramme de classe permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable. Nous avons pu construire le diagramme de classe indiqué ci-dessous ensuite le passage au modèle logique est effectué pour former la base de données de l'application.

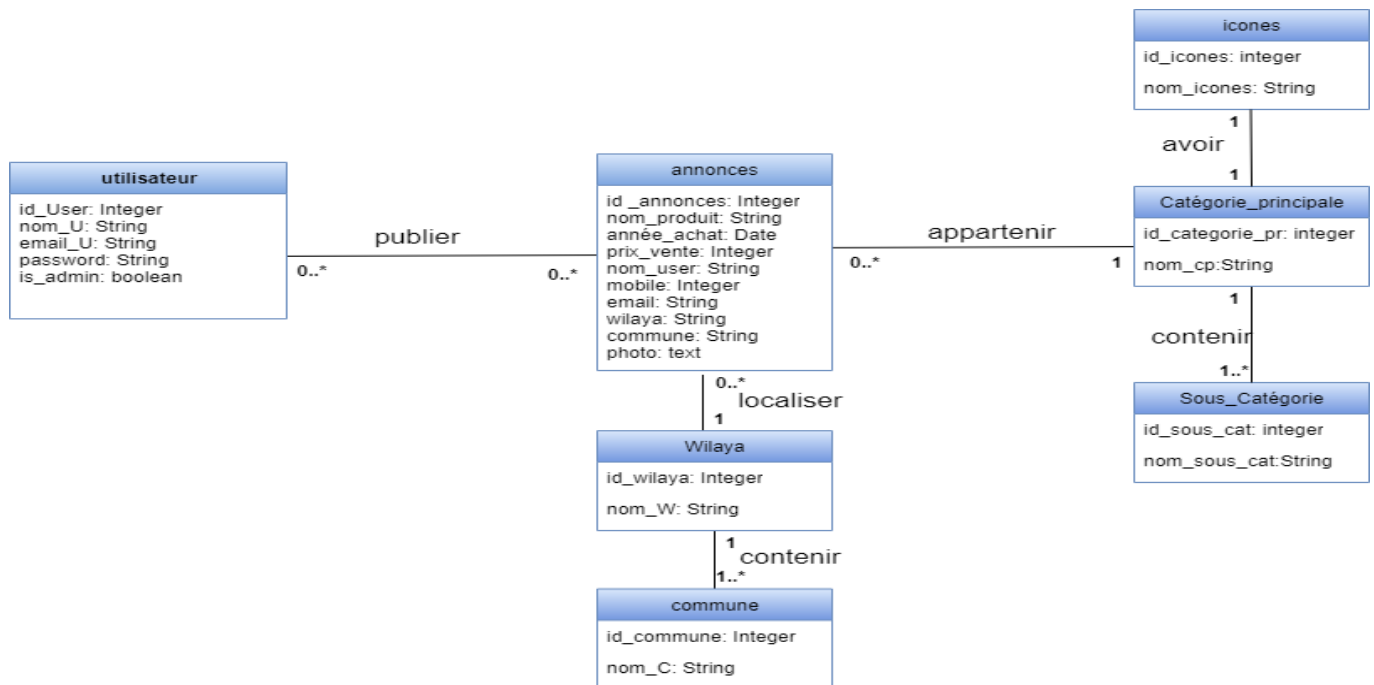


FIGURE 2.15 – Diagramme de classe

## 2.3.6 Le schéma relationnel

Pour réaliser ce passage, nous avons suivi des règles strictes et précises permettant de traduire le contenu conceptuel du diagramme de classe en modèle relationnel. Ces règles sont :

### 2.3.6.1 Règle de passage du diagramme de classe au modèle relationnel

Nous avons pu passer du diagramme de classes au modèle relationnel en se basant sur les règles suivantes :

**Règle 1 : Transformation des classes :**

Chaque classe devient une table. L'identifiant de la classe devient la clé primaire et chaque attribut de classe se transforme en un champ de table.

**Règle 2 : Association plusieurs-à-plusieurs :**

Toute association sous forme  $(?..*)$  à  $(?..*)$ , donne naissance à trois tables, dont la table dérivée de l'entité association possédera les clés primaires des deux autres tables comme clé primaire, plus ses propres attributs s'il s'agit d'une classe d'association.

**Règle 3 : Association d'un-à-plusieurs :**

Toute association binaire  $(?..1)$  à  $(?..*)$  donne naissance à une table dérivée de l'entité de la cardinalité  $(1..1)$ , et sa clé primaire est déposée comme clé étrangère dans la table dérivée de l'entité de la cardinalité  $(?..*)$ .

### 2.3.6.2 Le modèle relationnel

**Annonces** (id\_annonce, nom\_produit, année\_achat, prix\_vente, nom\_user, mobile, email, wilaya, commune, photos, #id\_catégorie\_pr, #id\_wilaya).

**Catégorie principale** (id\_catégorie\_pr, nom\_cp).

**Sous catégorie** (id\_sous\_cat, nom\_sous\_cat, #id\_catégorie\_pr).

**Icônes** (id\_icône, nom\_icône, #id\_catégorie\_pr).

**Wilaya** (id\_wilaya, nom\_w).

**Commune** (id\_commune, nom\_c, #id\_wilaya).

**Utilisateur** (id\_user, nom\_u, password, is\_admin).

**Publier** (#id\_user, #id\_annonce).

### 2.3.7 Conclusion

Dans ce chapitre, nous avons présenté le processus de développement UP, le langage de modélisation UML, et fait la modélisation des différents diagrammes : diagrammes de cas d'utilisations, diagrammes de séquences et diagrammes de classes. Cette étape de modélisation nous a permis d'avoir une vue générale sur le comportement théorique des fonctionnalités offertes par notre application. Cette base théorique nous servira de guide pour le développement de l'application, qui fera l'objet du chapitre suivant.



# Chapitre 3

## Réalisation

### 3.1 Introduction

Depuis le début de ce projet , nous avons bien déterminé les perspectives de l'application en commençant par une étude préalable, qui nous a permis d'avoir un objectif bien concret. Puis en arrivant à l'analyse, nous avons pu avoir une idée bien claire sur comment va être notre application web, et c'est ce qui nous a mené à mieux comprendre ses fonctionnalités.

Dans ce chapitre, nous allons nous appuyer sur la conception réalisée dans le chapitre précédent afin de présenter la concrétisation de l'application à travers son implémentation. Nous allons commencer par présenter les différents langages de développement ainsi que les outils de travail utilisés dans notre démarche, nous allons ensuite présenter en détail le principe de fonctionnement de ses outils, nous allons finir en présentant les interfaces des pages web de notre application ainsi que les différentes actions possibles à effectuer dans ces pages.

### 3.2 Langages et environnements de développement

#### 3.2.1 Le modèle MVC

Le MVC (M : Modèle, V : Vue, C : Contrôleur) est destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective.

Le pattern **MVC** permet de bien organiser son code source. Il va vous aider à savoir quels fichiers créer, mais surtout à définir leur rôle. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts.

- **Modèle** : cette partie gère les *données* de votre site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve donc entre autres les requêtes SQL.

- **Vue** : cette partie se concentre sur l’affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu’elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple une liste de messages.
- **Contrôleur** : cette partie gère la logique du code qui prend des décisions. C’est en quelque sorte l’intermédiaire entre le modèle et la vue. Le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le contrôleur contient exclusivement du PHP. C’est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d’accès) [9].



FIGURE 3.1 – Architecture MVC [9]

### 3.2.2 PHP

PHP signifie Personal Home Page, c’est un langage de script open source côté serveur utilisé pour créer des pages Web dynamiques. Les fichiers PHP peuvent contenir du texte, du HTML, du CSS, du JavaScript et du code PHP. Ce dernier est exécuté sur le serveur et le résultat est renvoyé au navigateur en format HTML. C’est probablement le langage de script le plus populaire.

PHP peut :

- Générer du contenu de page dynamique, créer, ouvrir, lire, écrire, supprimer et fermer des fichiers sur le serveur.
- Collecter des données de formulaire.
- Envoyer et recevoir des cookies.
- Ajouter, supprimer, modifier des données dans votre base de données.
- Être utilisé pour contrôler l’accès utilisateur.
- Chiffrer des données [10].

### 3.2.3 SQL

SQL (Structured Query Language) est un langage informatique standard pour la gestion de bases de données relationnelles et la manipulation de données. SQL est utilisé pour interroger, insérer, mettre à jour et modifier des données. La plupart des bases de données relationnelles prennent en charge SQL, ce qui constitue un avantage supplémentaire pour les administrateurs de

base de données, car elles sont souvent nécessaires pour prendre en charge des bases de données sur plusieurs plates-formes différentes [11].

### 3.2.4 HTML5

HTML (Hyper Text Markup Language) est un langage qui permet la description de documents, créée par **Tim Berners-Lee** suite à l'invention du web. Il est en mesure d'afficher des éléments textuels, graphiques et vidéo dans un navigateur web. Il est constitué d'un ensemble de balises et d'attributs utilisés pour gérer et organiser le contenu d'une page web [12].

### 3.2.5 CSS3

CSS (Cascading Style Sheets) est un langage complémentaire à celui du HTML5. Il se charge de la mise en forme et de la mise en page des éléments. Il permet de gérer l'apparence de la page tout en précisant les couleurs, le positionnement des objets, des marges, etc [12].

### 3.2.6 Java Script

JavaScript (JS) est un langage de script, principalement utilisé sur le Web. Il est utilisé pour améliorer les pages HTML et est généralement intégré au code HTML. C'est un langage interprété, ce qui signifie qu'il n'a pas besoin d'être compilé. Il rend les pages Web de manière interactive et dynamique, cela permet aux pages de réagir aux événements, d'afficher des effets spéciaux, d'accepter des textes variables, de valider des données, de créer des cookies, de détecter le navigateur de l'utilisateur, etc [13].

### 3.2.7 jQuery

jQuery est une bibliothèque JavaScript rapide, petite et riche en fonctionnalités. Il simplifie considérablement la traversée et la manipulation de documents HTML, la gestion des événements, l'animation et Ajax grâce à une API facile à utiliser qui fonctionne sur une multitude de navigateurs. Avec une combinaison de polyvalence et d'extensibilité, jQuery a changé la façon dont des millions de personnes écrivent du JavaScript [14].

### 3.2.8 Ajax

AJAX (Asynchronous JavaScript and XML) ce qui, transcrit en français, signifie « JavaScript et XML asynchrones ».

Derrière ce nom se cache un ensemble de technologies destinées à réaliser de rapides mises à jour du contenu d'une page Web, sans qu'elles nécessitent le moindre rechargement visible par l'utilisateur de la page Web. Les technologies employées sont diverses et dépendent du type de requêtes que l'on souhaite utiliser, mais d'une manière générale le JavaScript est constamment présent.

Ajax est une technologie JavaScript fort répandue qui permet d'envoyer des requêtes au serveur et de recevoir des réponses sans rechargement de la page. Il est par ce moyen possible de modifier dynamiquement le DOM (Document Object Model), donc une partie de la page [15].

## 3.3 Outils de développement

### 3.3.1 XAMPP

XAMPP est l'environnement de développement PHP le plus populaire. XAMPP est une distribution Apache entièrement gratuite et facile à installer contenant MySQL, PHP et Perl. Le paquetage open source XAMPP a été mis au point pour être incroyablement facile à installer et à utiliser [16].

### 3.3.2 MYSQL

C'est un système de gestion de base de données relationnelle (SGBDR) open source avec un modèle client-serveur. Il est développé, distribué et pris en charge par Oracle Corporation.

### 3.3.3 PHPStorm

Jetbrains PHPStorm est un environnement de développement intégré spécialisé pour le développement Web. Initialement conçu pour le développement PHP avec tout le support de Webstorm, grâce à son extensibilité, plusieurs plugins existent pour étendre ses fonctionnalités. Il supporte nativement les langages PHP, HTML, CSS et JavaScript.

Si plusieurs développeurs Web se penchent vers cet IDE, c'est grâce aux nombreux avantages et fonctionnalités que possède PHPStorm qui rendent vraiment facile la vie des développeurs en occurrence :

1. Le support de la coloration syntaxique.
2. Un générateur de code.
3. L'auto compilation.
4. L'auto complétion intelligent du code.
5. Une documentation du langage PHP intégré.
6. La factorisation du code.
7. File watcher, un outil qui fait la surveillance des fichiers surtout pour la transpilation des codes pour ceux qui utilisent des préprocesseurs comme SASS ou Webpack.
8. Un support de gestion de version (GIT, SVN) intégré.
9. Les supports des framework (Symfony) PHP intégré.
10. Les supports des moteurs de templates comme blade, twig intégré [17].

### 3.3.4 Draw.io

C'est une application gratuite en ligne, accessible via le navigateur qui permet de dessiner des diagrammes ou des organigrammes. Cet outil nous propose de concevoir toutes sortes de diagrammes, de dessins vectoriels, de les enregistrer ou de les exporter en format image, XML, PDF, etc [18].

### 3.3.5 Photoshop

Logiciel de traitement et de retouche d'images et de photos produit par la société Adobe. Photoshop est devenu le standard en matière de gestion des images matricielles (ou images "bitmap", constituées d'un "tapis de points"). Un logiciel tel qu'Illustrator, lui, gère l'image numérique sous la forme de vecteurs (on parle alors d'images vectorielles) [19].

## 3.4 Frameworks utilisés

Un Framework est, comme son nom l'indique en anglais, un « cadre de travail ». L'objectif d'un Framework est généralement de simplifier le travail des développeurs informatiques, en leur offrant une architecture "prête à l'emploi" et qui leur permette de ne pas repartir de zéro à chaque nouveau projet.

Les Frameworks sont comparables aux patrons de couture. Les principaux avantages sont donc :

- La réutilisation des codes.
- La standardisation de la programmation.
- La formalisation d'une architecture adaptée aux besoins de chaque entreprise.

À noter aussi que les Frameworks sont toujours « enrichis » de l'expérience de tous les développements antérieurs.

En bon français, plusieurs termes et expressions permettent actuellement de désigner le Framework : cadriciel, canevas, socle d'applications, cadre d'application, environnement de développement, boîte à outils, etc [20].

Nous avons utilisé deux Frameworks :

### 3.4.1 Bootstrap

C'est un Framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement. Ce Framework est pensé pour développer des sites avec un DESIGN RESPONSIVE, qui s'adapte à tout type d'écran, et en priorité pour les smartphones (mobile first). Il fournit des outils avec des styles déjà en place pour des typographies,

des boutons, des interfaces de navigation et bien d'autres encore. On appelle ce type de framework un "Front-End Framework" [21].

### 3.4.2 Laravel

- **Définition de laravel**

C'est un Framework PHP open source qui suit la logique du design pattern MVC et qui propose des outils pour construire une application web. Le créateur de Laravel, Taylor Otwell, a regroupé les bibliothèques les plus performantes pour effectuer chaque fonctionnalité nécessaire à la création d'un site web. Laravel comprend notamment des fonctionnalités exclusives qui sont propres à lui comme le moteur de template Blade (il s'occupe de séparer tout le code de présentation, tout ce qui est HTML et CSS, donc la VUE du modèle MVC, et le code applicatif c'est à dire le code PHP) [22].

Laravel requiert l'installation d'un gestionnaire de dépendances « composer ».

Pour installer laravel nous devons écrire dans le composer la commande suivante :

```
composer create-project --prefer-dist laravel/laravel NomDuProjet
```

Laravel permet de paramétrer l'authentification avec la commande « php artisan make:auth ». Cette commande va créer toutes les routes et toutes les classes nécessaires pour l'authentification.

Le Framework Bootstrap est intégré dans laravel.

- **Organisation de laravel**

Laravel est composé d'une hiérarchie de dossiers et fichiers représentée dans cette figure :

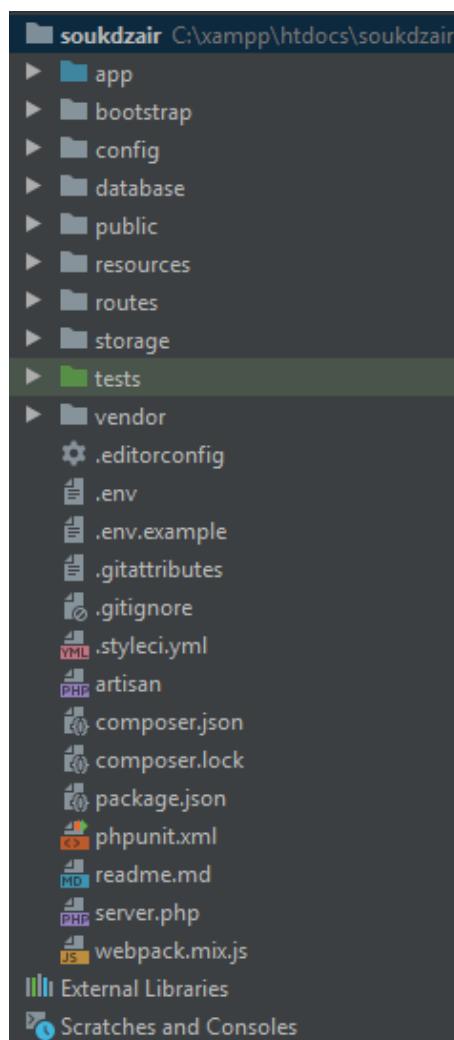


FIGURE 3.2 – Organisation de laravel

- app** : ce dossier contient les éléments essentiels de l'application : les **Models**, les **Providers** (tous les fournisseurs de services, servent à initialiser les composants), le dossier **http** (tout ce qui concerne la communication : contrôleurs, middlewares, et requêtes).
- bootstrap** : scripts d'initialisation de laravel pour le chargement automatique des classes, la fixation de l'environnement et des chemins, et pour le démarrage de l'application.
- config** : toutes les configurations : application, authentification, cache, base de données, espaces de noms, emails, systèmes de fichier, session...
- database** : les migrations et les populations.
- public** : tout ce qui doit apparaître dans le dossier public du site : images, CSS, scripts...
- ressources** : vues, fichiers de langage et assets (par exemple les fichiers LESS ou Sass).
- routes** : les chemins de l'application sont définis dans le fichier web.php qui se trouve dans le dossier routes.
- storage** : données temporaires de l'application : vues compilées, caches, clés de session...

- tests** : fichiers de tests unitaires.
- vendor** : tous les composants de laravel et de ses dépendances.
- artisan** : outil en ligne de laravel pour des tâches de gestion.
- composer.json** : fichier de référence de « Composer ».
- phpunit.xml** : fichier de configuration de phpunit (pour les tests unitaires).
- env** : fichier pour spécifier l'environnement d'exécution.

- **Controllers et Middleware**

La grande majorité des actions qu'un utilisateur enclenche mènera à l'appel d'une méthode d'un contrôleur. Il existe deux types de contrôleurs :

- Le contrôleur classique : dont les fonctions sont créées par le développeur.
- Le contrôleur de type ressource : contient sept fonctions prédéfinies (index, create, edit, store, show, update et destroy).

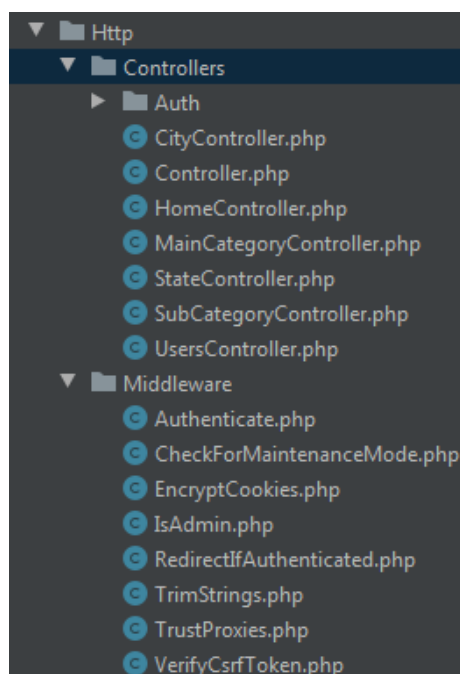


FIGURE 3.3 – Vue sur les Contrôleurs et Middleware

Une fois le projet ouvert, les différents Controllers et Middlewares sont contenus dans le dossier « Http ».

Les « Controllers » vont permettre de gérer les différentes actions demandées par les utilisateurs lors d'une action.

Pour créer un contrôleur avec le Framework laravel il faut utiliser la commande suivante :

```
php artisan make :controller nom_du_controller --resource
```



Le paramètre « `-resource` » permet d'ajouter les différentes méthodes par défaut du Framework laravel :

- La méthode « `index` » : permet d'afficher une liste ressource.
- La méthode « `create` » : pour créer des nouveaux éléments dans la base de données via un formulaire.
- La méthode « `store` » : pour stocker les éléments dans la base de données.
- La méthode « `show` » : permet d'afficher une ressource spécifique.
- La méthode « `edit` » : permet de modifier une ressource spécifique.
- La méthode « `update` » : permet de mettre à jour une ressource spécifique.
- La méthode « `destroy` » : permet de supprimer une ressource spécifique.

Les « Middlewares » permettent de filtrer différentes requêtes http.

Pour créer un middleware avec le Framework laravel il faut utiliser la commande suivante :

```
php artisan make :middleware nom_du_middleware
```

Les middlewares et les contrôleurs seront directement créés dans leurs dossiers respectifs.

#### • Les migrations (le model)

La notion de « Migrations » dans le Framework laravel : une migration permet de créer et de mettre à jour un schéma de base de données. Autrement dit nous pouvons créer des tables, des colonnes dans ces tables, en supprimer, créer des index, etc. Tout ce qui concerne la maintenance de nos tables peut être pris en charge par cet outil. En utilisant la syntaxe PHP au lieu de SQL.

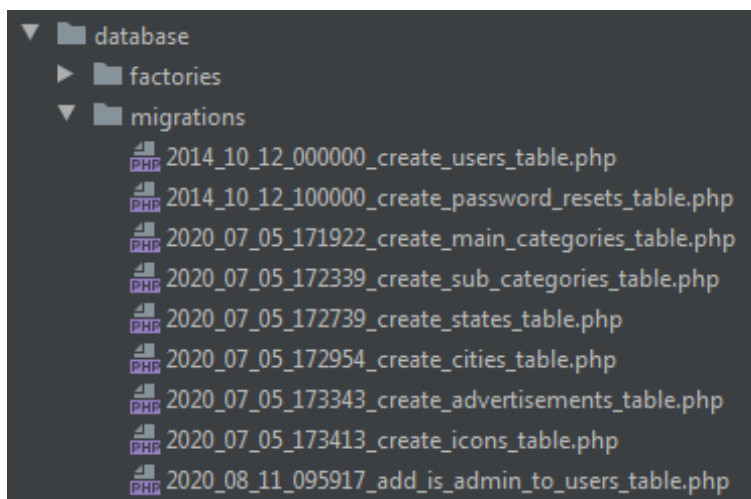


FIGURE 3.4 – Le dossier database

Les différentes migrations se situent dans le dossier « `database` » > « `migrations` ». Les migrations permettent de gérer les différentes colonnes de notre base de données. Par défaut, il y a 2 migrations déjà mises en place : « `create_users_table` » et « `create_password_resets_table` ».

Pour chaque migration, un model doit être créé dans le dossier « app ». Ici, le model « User » est déjà créé par défaut.

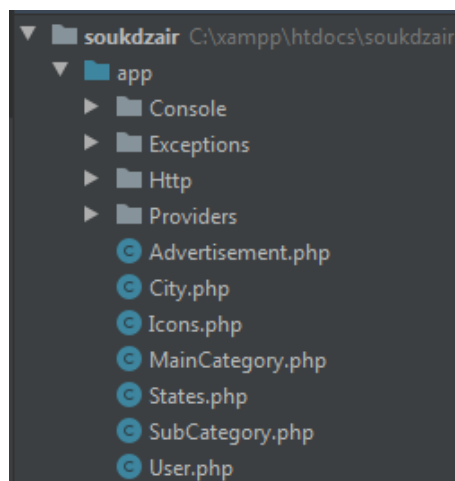


FIGURE 3.5 – Les différents models qui se trouvent dans app dont le model user

Pour créer une nouvelle migration, il faut utiliser la commande suivante :

```
php artisan make :migration nom_migration
```

- **Vues (view)**

Pour afficher correctement une page Web dans le navigateur, il faut utiliser une vue. C'est la vue qui est en charge de générer le code HTML. En plus des balises HTML, elle utilise des directives et instructions que le moteur d'affichage Blade met à sa disposition. Les vues sous laravel sont placées dans le dossier **resources/views**.

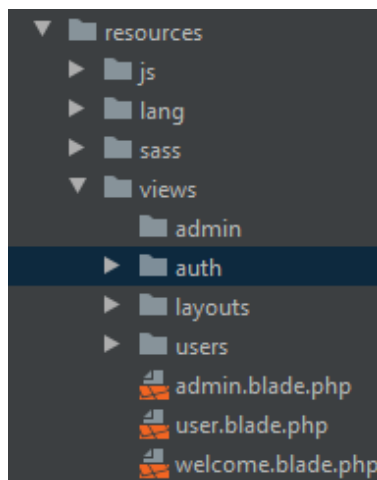


FIGURE 3.6 – Le dossier ressources

Les différentes vues de l'application se trouvent dans le dossier « resources » > « views ».

Le Framework laravel fournit un « template » nommé « Blade » pour l'utilisation des vues.

Pour créer une nouvelle vue, faites un clic droit sur le dossier « views ». Allez sur « New > » et choisissez « PHP File ». Lorsque vous devez choisir le nom du fichier, vous devez utiliser la formulation suivante : `nom_vue.blade.php`.

- **Routes**

Le système de routing permet de créer des liens. Lorsqu'on crée des liens, on utilise juste les noms des routes et laravel va automatiquement insérer l'URL correct.

Les chemins de notre application sont définis dans le fichier `web.php` dans le dossier routes représenté dans l'image suivante :

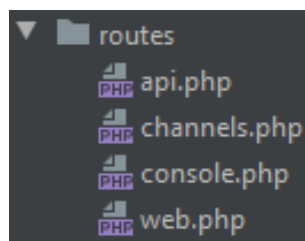


FIGURE 3.7 – Le dossier routes

Les routes utilisées pour l'application web se trouvent généralement dans le dossier « routes » et dans le fichier « web.php ». Elles permettent de naviguer entre les différentes pages de notre application.

Le Framework laravel utilise différents types de routes :

- GET** : ce type est utilisé pour récupérer des éléments.
- POST** : Ce type est utilisé pour poster des nouveaux éléments via un formulaire notamment.
- DELETE** : ce type est utilisé pour supprimer un ou plusieurs éléments.
- PUT** : Ce type est utilisé pour modifier un élément depuis un formulaire.

## 3.5 Présentation des interfaces graphiques

Nous allons présenter dans ce qui suit, les imprimes-écran des principales interfaces réalisées pour notre application web.

### 3.5.1 La page d'accueil

C'est la page d'accueil qui s'affiche dès l'accès à notre application web. Nous pouvons à travers cette page :

- Voir les annonces.
- S'authentifier ou bien s'inscrire ou déposer une annonce.

- La recherche d'une annonce selon le nom du produit ou bien selon la combinaison de la wilaya et de la catégorie.

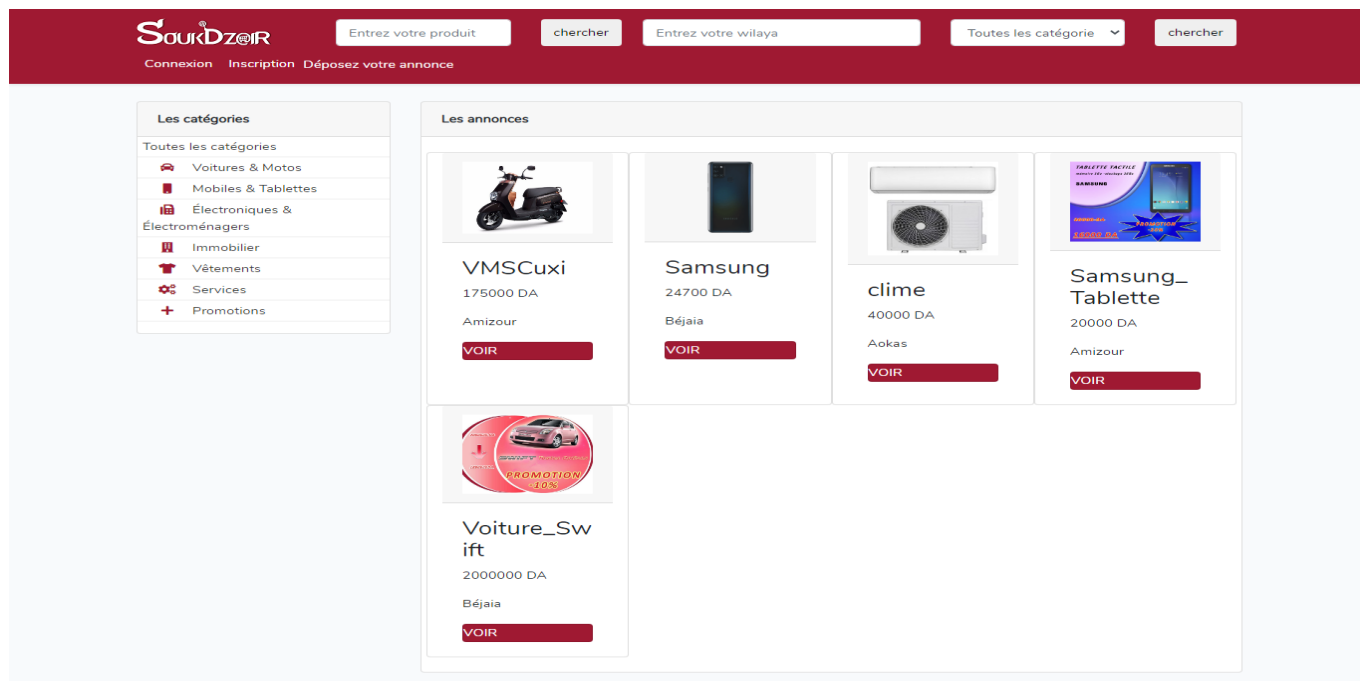


FIGURE 3.8 – La page d'accueil

### 3.5.2 Page de pré-inscription

Notre application web met à la disposition de ses visiteurs un formulaire d'inscription accessible à partir du menu inscription dans la barre des menus en haut de la page d'accueil.

The image shows the registration page of the SoukDz@R website. The header is dark red and contains the logo 'SoukDz@R' on the left. To the right of the logo are four search-related input fields: 'Entrez votre produit' with a 'chercher' button, 'Entrez votre wilaya' with a 'chercher' button, 'Toutes les catégorie' with a dropdown arrow, and another 'chercher' button. Below the header, there are links for 'Connexion', 'Inscription', and 'Déposez votre annonce'. The main content area is light blue and features a white registration form titled 'Inscription'. The form contains four input fields: 'Nom', 'Adresse e-mail', 'Mot de passe', and 'Confirmer le mot de passe'. A blue button labeled 'S'inscrire' is positioned at the bottom of the form.

FIGURE 3.9 – La page de pré-inscription

### 3.5.3 Page d'authentification

Après la phase d'inscription présentée dans la figure 3.9, l'utilisateur doit s'authentifier afin d'avoir accès à la publication d'annonces. Pour cela, il doit introduire son adresse-mail et son mot de passe.

Connexion

Adresse e-mail

Mot de passe

Souviens-toi de moi

S'identifier [Mot de passe oublié?](#)

FIGURE 3.10 – La page d'authentification

### 3.5.4 Les interfaces de publications d'annonces

Après s'être authentifié, l'utilisateur peut alors sélectionner la catégorie de l'annonce à ajouter en y introduisant toutes les informations nécessaires.

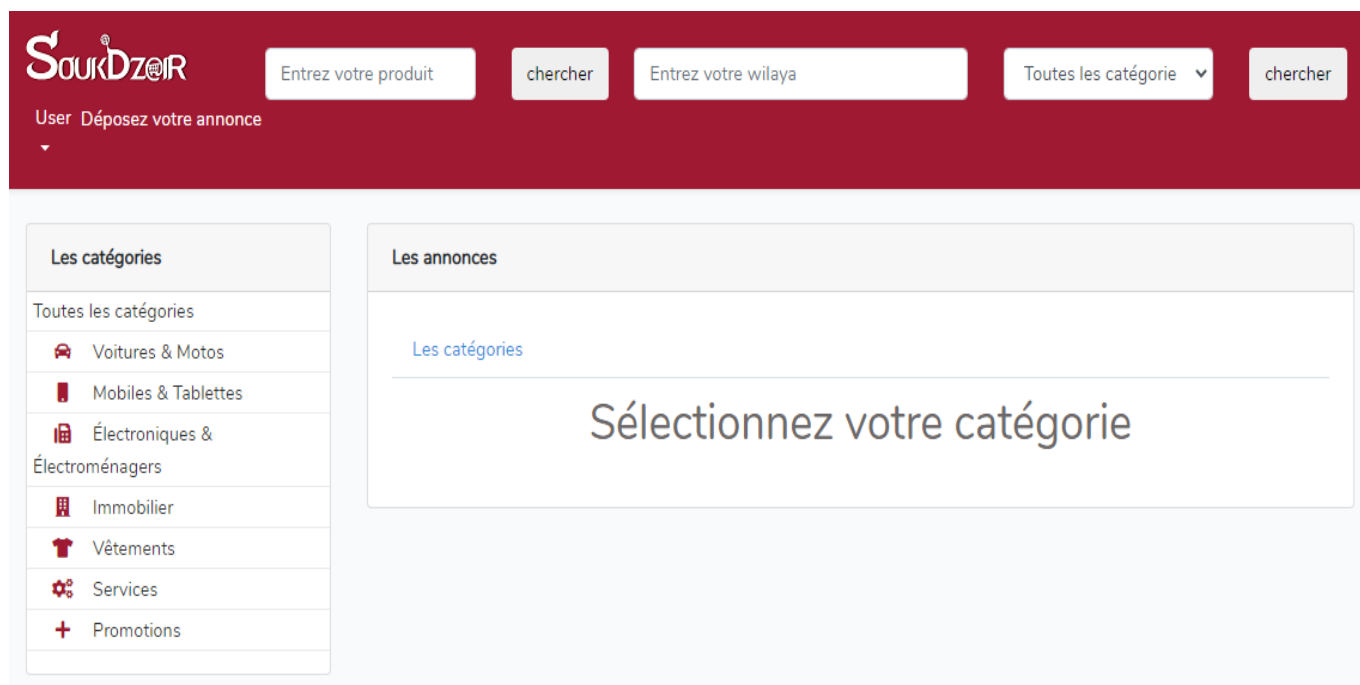


FIGURE 3.11 – Interface du choix de la catégorie

- Interface de publication d'annonce

The screenshot shows the 'SoukDz@R' advertisement posting interface. At the top, there is a dark red header with the logo and search fields: 'Entrez votre produit', 'chercher', 'Entrez votre wilaya', 'Toutes les catégorie', and 'chercher'. Below the header, the interface is divided into two main sections: 'Les catégories' on the left and 'Les annonces' on the right.

**Les catégories** sidebar includes: Toutes les catégories, Voitures & Motos, Mobiles & Tablettes, Électroniques & Électroménagers, Immobilier, Vêtements, Services, and Promotions.

**Les annonces** section is titled 'Vêtements' and contains the following fields:

- \*Champs obligatoires**
- Sélectionnez une sous catégories \***: A dropdown menu with 'Select' as the current selection.
- Nom du produit \***: A text input field containing 'nom du produit'.
- L'année d'achat \***: A text input field containing 'L'année d'achat'.
- Prix de vente prévu \***: A text input field containing 'Prix de vente prévu'.
- Votre nom \***: A text input field containing 'Votre nom'.
- Votre numéro de téléphone \***: A text input field containing 'Votre numéro de téléphone'.
- Votre adresse mail \***: A text input field containing 'Votre adresse mail'.
- Votre wilaya \***: A dropdown menu with 'Select' as the current selection.
- Commune \***: A text input field containing 'Entrez votre commune'.
- Vos photos \***: A file upload area with a 'Sélect. fichiers' button and the text 'Aucun fichier choisi'.

At the bottom of the form, there are two buttons: 'Publier votre annonce' (blue) and 'Réinitialiser' (grey).

FIGURE 3.12 – Interface de publication d'annonce

- Les interfaces de recherche d'annonces

- Selon le nom du produit : cette interface permet la recherche des annonces selon un produit spécifique.



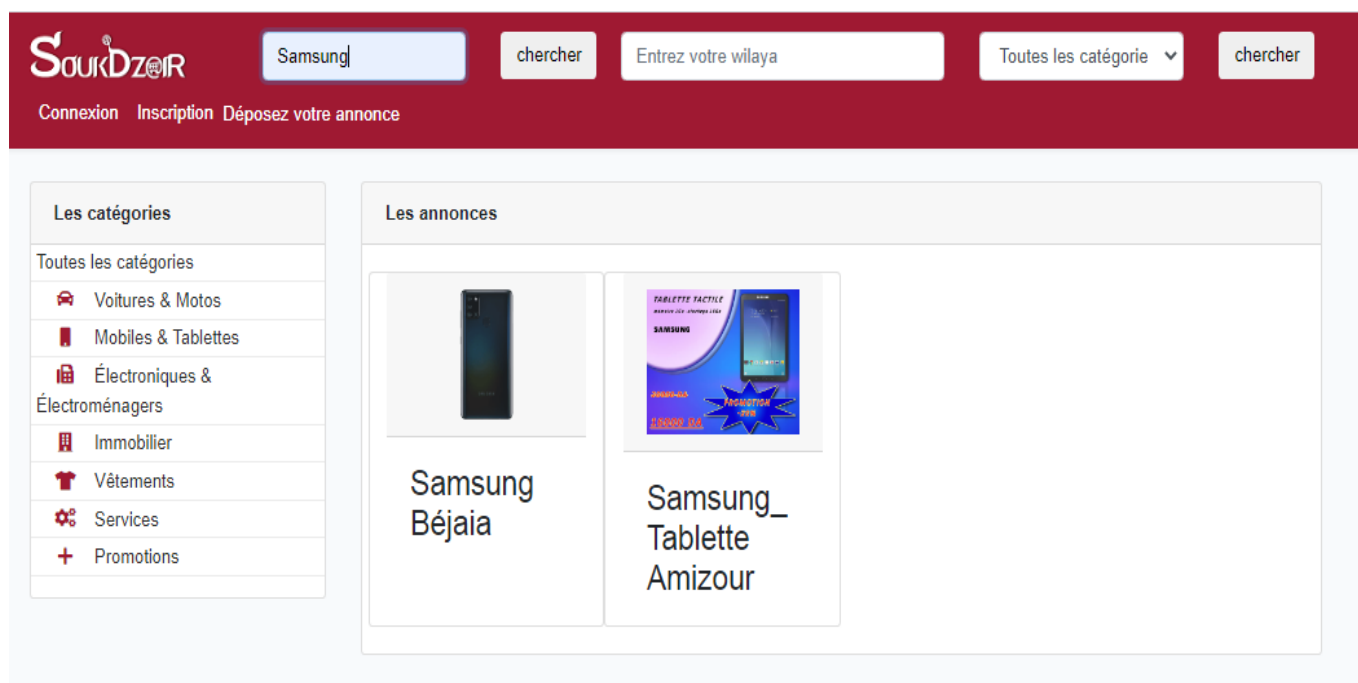


FIGURE 3.13 – Interface de recherche d’annonce selon un produit

- Selon la combinaison de la wilaya et la catégorie : cette interface permet la recherche des annonces selon la wilaya et la catégorie.

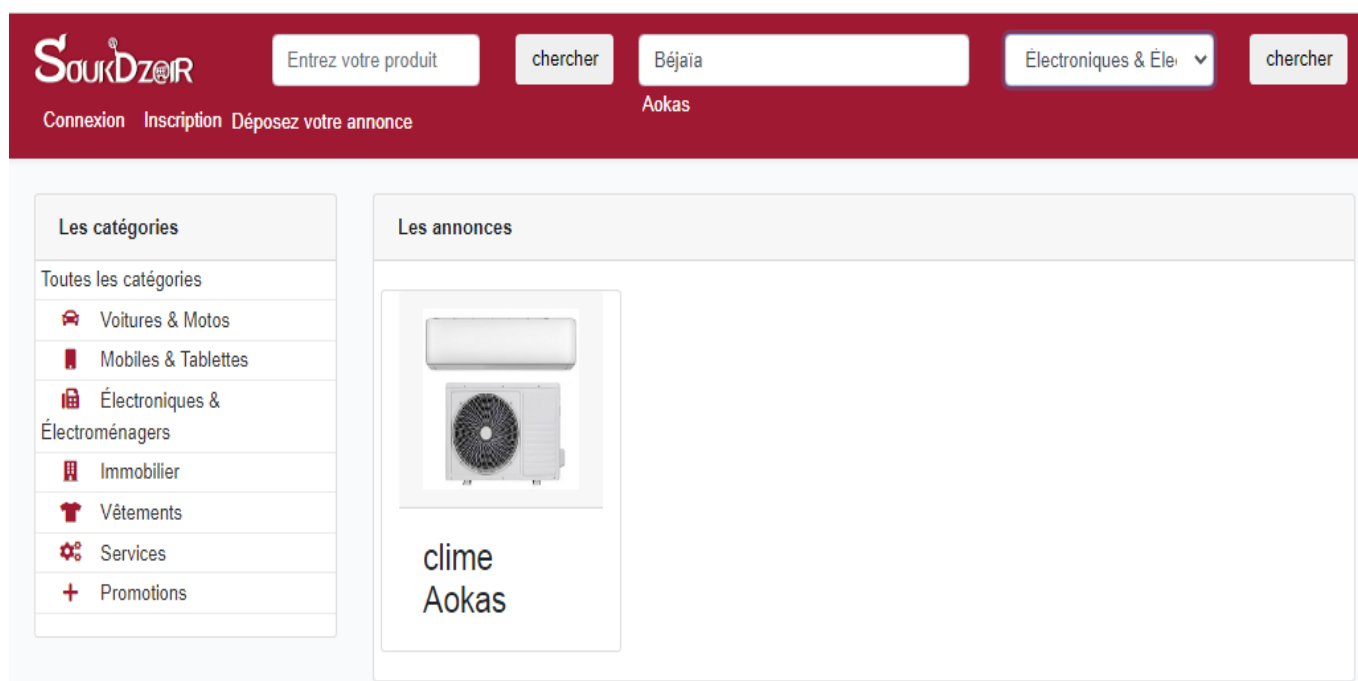


FIGURE 3.14 – Interface de recherche d’annonce selon la wilaya et la catégorie

### 3.5.5 Les interfaces montrant l’affichage de l’annonce selon sa catégorie

□ Exemple de catégorie « Promotion ».

The screenshot displays the SoukDzair website interface. At the top, there is a dark red navigation bar with the logo 'SoukDzair' on the left and search fields on the right. The search fields include 'Entrez votre produit', 'chercher', 'Entrez votre wilaya', 'Toutes les catégorie', and another 'chercher'. Below the navigation bar, there are links for 'Connexion', 'Inscription', and 'Déposez votre annonce'. The main content area is divided into two sections: 'Les catégories' on the left and 'Les annonces' on the right. The 'Les catégories' section lists various categories, with 'Promotions' highlighted in a dark red bar. The 'Les annonces' section displays two promotional ads. The first ad is for a 'Samsung\_Tablette' priced at '20000DA' in 'Amizour'. The second ad is for a 'Voiture\_Swift' priced at '2000000 DA' in 'Béjaia'. Both ads feature images of the respective products and a 'PROMOTION 10%' badge.

FIGURE 3.15 – Les annonces de la catégorie promotions

□ Exemple de catégorie « vêtements ».

The screenshot shows the SoukDzair website interface. At the top, there is a dark red header with the logo 'SoukDzair' on the left. To the right of the logo are two search bars: 'Entrez votre produit' with a 'chercher' button, and 'Entrez votre wilaya' with a 'chercher' button. Below the search bars are links for 'Connexion', 'Inscription', and 'Déposez votre annonce'. The main content area is divided into two columns. The left column, titled 'Les catégories', lists various product categories: 'Toutes les catégories', 'Voitures & Motos', 'Mobiles & Tablettes', 'Électroniques & Électroménagers', 'Immobilier', 'Vêtements' (highlighted in red), 'Services', and 'Promotions'. The right column, titled 'Les annonces', displays a single listing for a 'T-shirt Amizour'. The listing includes a thumbnail image of a white t-shirt with a black graphic of an evolutionary sequence (from an ape to a human), the title 'T-shirt Amizour', and a small red eye icon below the title.

FIGURE 3.16 – Les annonces de la catégorie vêtements

- Voir plus sur l'annonce : après avoir cliquer sur l'œil on obtient des détails sur l'annonce.

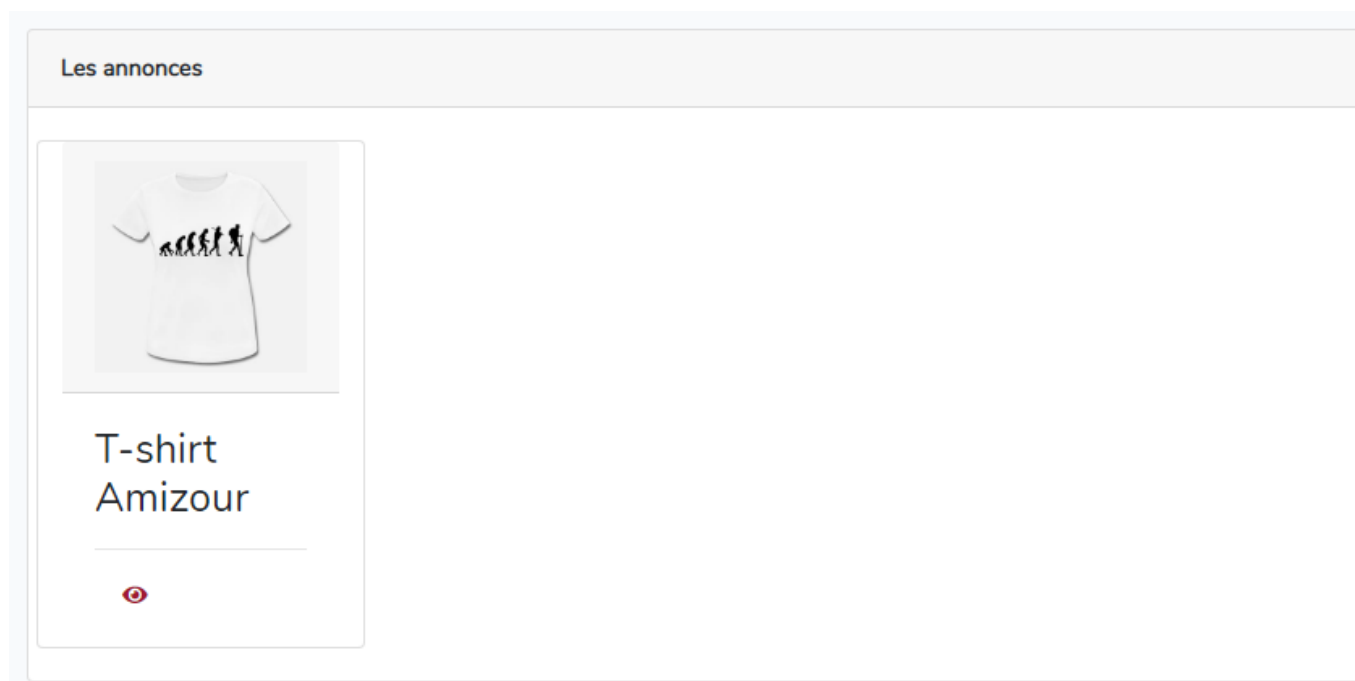


FIGURE 3.17 – Interface montrant les détails de l'annonce après avoir cliquer sur l'icône sous forme d'œil



FIGURE 3.18 – Les détails sur l'annonce

### 3.5.6 Page représentant les activités de l'administrateur

- Suppression de l'annonce : l'administrateur peut supprimer les différentes annonces publiées par les annonceurs. En effet, l'administrateur choisit l'annonce qu'il souhaite supprimer dans le menu "**des catégories**", puis un simple clique sur la corbeille l'annonce sera supprimée, puis retour à la page d'accueil.

- **Exemple de suppression d'un produit**

Nous allons prendre comme exemple le produit « Samsung » de la catégorie « Mobiles & Tablettes ».

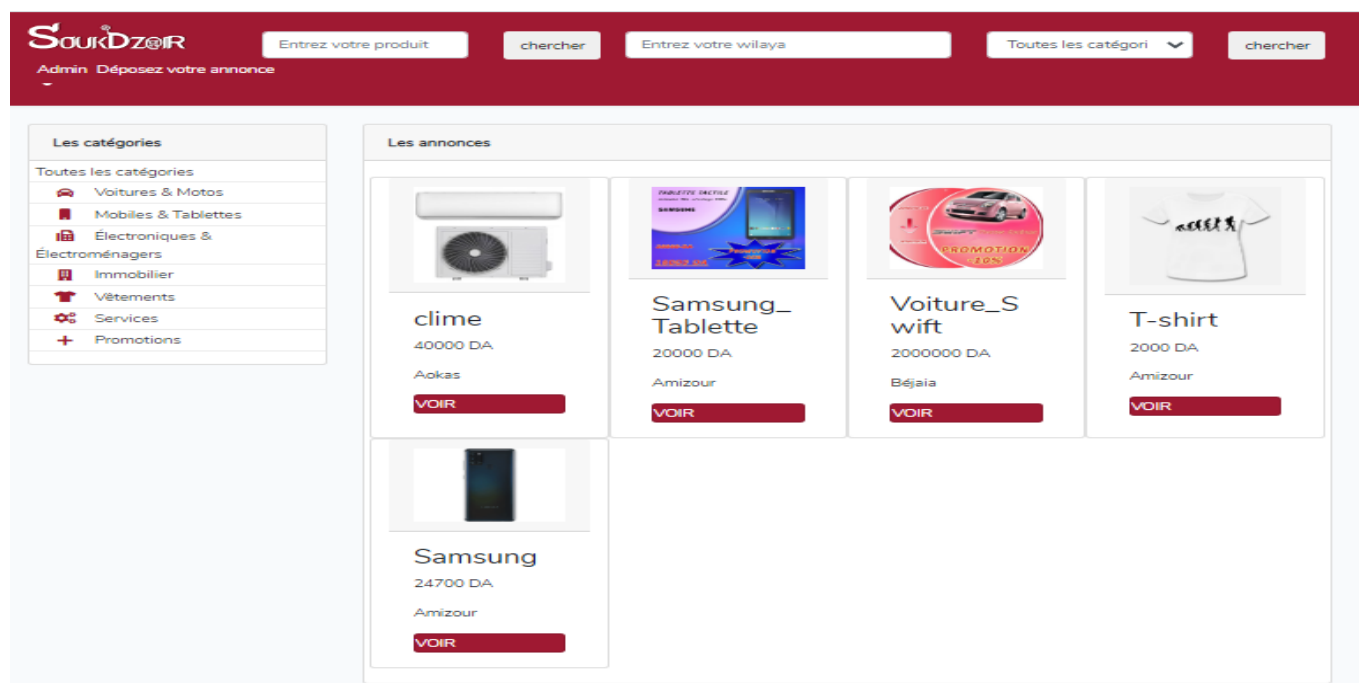


FIGURE 3.19 – L'interface d'accueil avant la suppression du produit « Samsung »

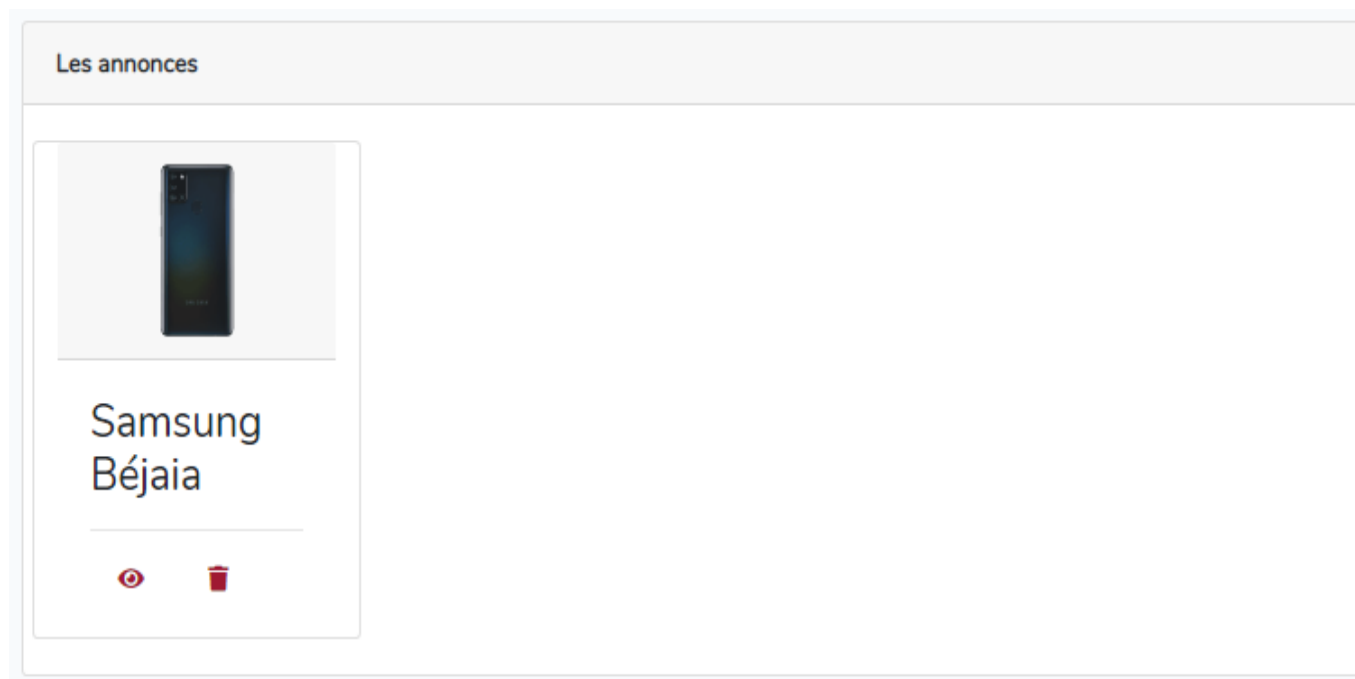


FIGURE 3.20 – Interface montrant la suppression de l’annonce après avoir cliquer sur la corbeille

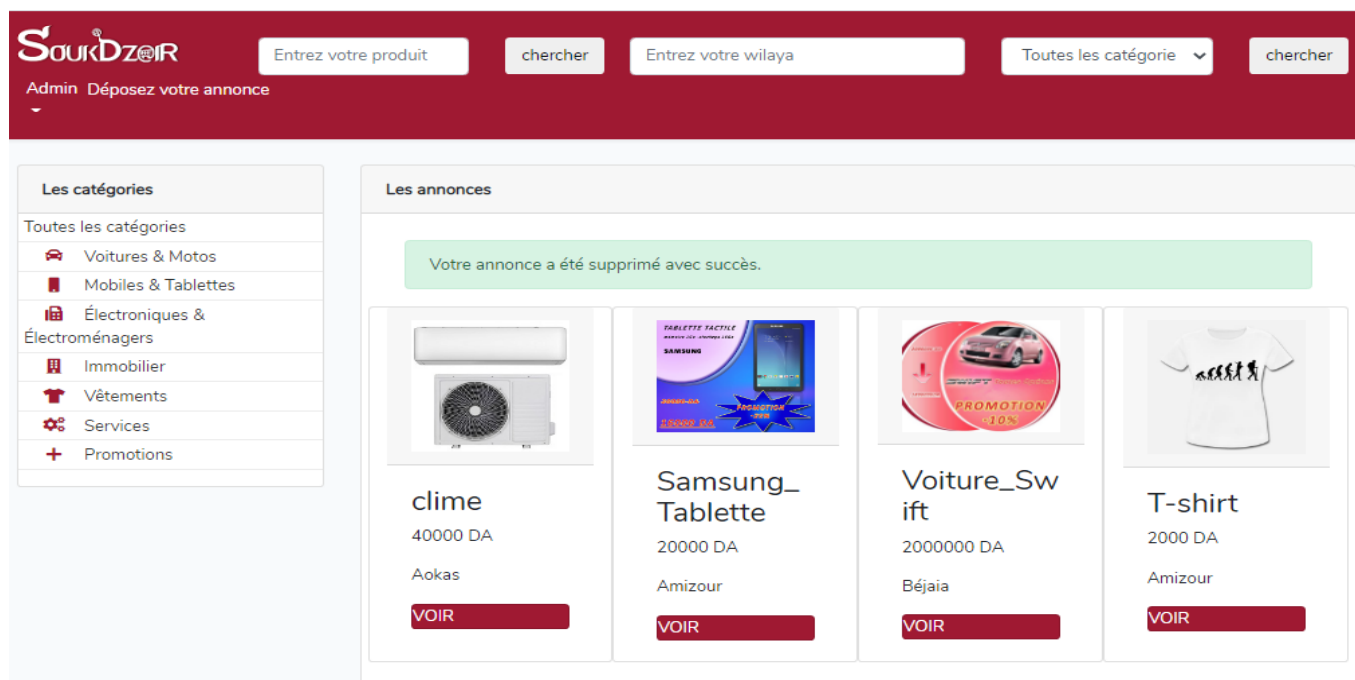


FIGURE 3.21 – Interface montrant que l’annonce a été bien supprimée

## 3.6 Conclusion

Au cours de ce chapitre, nous avons présenté les environnements de développements ainsi que les langages de programmation dont nous nous sommes servis afin de réaliser notre application, notamment l'outil Laravel qui nous a fourni plusieurs avantages dont sa documentation qui est excellente et complète, ce qui nous a donné la possibilité de trouver rapidement des réponses à nos questions, ainsi que sa contribution à sécuriser l'application Web en la protégeant contre les plus graves failles de sécurité. Grâce à cet outil nous avons pu avoir une application mieux structurée et mieux sécurisée, ce qui nous a permis d'atteindre nos objectifs initiaux décrits dans les étapes précédentes et à aboutir à un produit utilisable qui répond non seulement aux exigences initiales mais qui est aussi apte pour une quelconque mise à jour si de nouvelles exigences se présentent.

Pour terminer ce chapitre, nous avons présenté les principales interfaces graphiques de SoukDzair pour bien illustrer le travail qui a été réalisé.

# Conclusion générale

Au cours de ce travail, nous avons travaillé dans le but de concevoir et de réaliser une plateforme de gestion de différents types d'annonces, nous sommes parvenus à réaliser cette plateforme web en suivant les différentes étapes du processus UP.

Nous avons commencé par une étude approfondie, dans le but d'identifier les problèmes rencontrés au quotidien par les entreprises et les particuliers, pour en extraire les besoins fonctionnels de l'application. Nous avons ensuite pris connaissance des acteurs essentiels de ce système.

Ensuite, nous avons montré la phase de conception en traçant les principaux diagrammes UML répondant aux attentes de nos besoins. Après application des règles de passage du diagramme de classes vers le modèle relationnel nous avons réussi à implémenter notre base de données.

Enfin, nous avons réalisé notre application web, et traité les phases nécessaires à la réalisation, où nous avons appris à manipuler le Framework Laravel, et approfondi nos connaissances sur le langage PHP, HTML, Ajax, CSS, MySQL et Bootstrap.

Ce projet nous a été très bénéfique, il nous a permis de nous rendre compte de la difficulté et de l'importance de la réalisation d'une bonne étude préalable, et d'appliquer les connaissances théoriques que nous avons acquises tout au long de notre cursus. En effet, il nous a aussi permis d'enrichir nos connaissances théoriques et pratiques dans le domaine de la conception et de la programmation.

En perspectives, nous envisageons d'améliorer et d'enrichir cette application par les fonctionnalités suivantes :

- Ajouter d'autres langues pour l'interface de l'application.
- Intégrer les interactions des utilisateurs sur les annonces (commentaires, notes et réactions).
- Ajouter la géolocalisation afin de permettre aux clients de retrouver les annonceurs les plus proches ainsi que les annonces publiées au alentour de leur localisation géographique.
- Améliorer les interfaces grâce à l'intégration des thèmes.



# Bibliographie

- [1] Y.Brahimi, Conception et réalisation d'une application web pour la gestion d'un jardin d'enfants, Mémoire de Licence, Université Abou Bekr Belkaid, Tlemcen, 2015.
- [2] Z.Echaouch, A.Hadyaoui, Développement d'un site dynamique d'annonce gratuit, Université du Sousse, [https://www.memoireonline.com/03/12/5480/m\\_Developpement-d-un-site-dynamique-d-annonce-gratuit-0.html](https://www.memoireonline.com/03/12/5480/m_Developpement-d-un-site-dynamique-d-annonce-gratuit-0.html), 2010, Consulté le 30/08/2020.
- [3] | ESSABBAR DRISS, Processus Unifié 03/2019.
- [4] Sophnouille, UP : Unified Process, Publié le 4 mai 2004 - Mis à jour le 19 juin 2020.
- [5] Grandy Booch, le guide d'utilisation en UML, édition 2002.
- [6] Processus Unifié UP & 2TUP | Saad Mansouri - Academia.edu 27/03/2019.
- [7] Eskandar Kouicem, Azza Dridi, Med Nadir Boukelal, Université de Constantine 2 - Licence en informatique 2016.
- [8] Christian Soutou, Concevoir une base de données grâce à UML 2, 15 mars 2007.
- [9] <https://openclassrooms.com/fr/courses/4670706-adoptez-une-architecture-mvc-en-php/4678736-comment-fonctionne-une-architecture-mvc> (Consulté le 05/08/2020).
- [10] [https://www.w3schools.com/php/php\\_intro.asp](https://www.w3schools.com/php/php_intro.asp) (Consulté le 05/08/2020).
- [11] <https://www.businessnewsdaily.com/5804-what-is-sql.html> (Consulté le 05/08/2020).
- [12] Michel Martin, HTML5 & CSS3, 2ème édition Pearson France, 2013,(Consulté le 05/08/2020).
- [13] <https://www.techopedia.com/definition/3929/javascript-js> (Consulté le 05/08/2020).
- [14] <https://jquery.com/> (Consulté le 30/08/2020).
- [15] <https://openclassrooms.com/fr/courses/3613341-decouvrez-le-framework-php-laravel/3619425-ajax> (Consulté le 06/08/2020).
- [16] <https://www.apachefriends.org/fr/index.html> (Consulté le 30/08/2020).
- [17] <https://informagenie.com/2037/ide-pour-php-phpstorm/> (Consulté le 06/08/2020).
- [18] <https://www.tice-education.fr/index.php/tous-les-articles-et-ressources/articles-internet/819-draw-io-un-outil-pour-dessiner-des-diagrammes-en-ligne> (Consulté le 06/08/2020).
- [19] <http://www.mosaique-info.fr/glossaire-web-referencement-infographie-multimedia-informatique/p-glossaire-informatique-et-multimedia/228-photoshop-definition.html> (Consulté le 06/08/2020).

- [20] [https : //www.1min30.com/dictionnaire – du – web/framework](https://www.1min30.com/dictionnaire-du-web/framework) (Consulté le 06/08/2020).
- [21] [https : //www.journaldunet.com/web–tech/developpeur/1159810–bootstrap–definition–tutoriels – astuces – pratiques/](https://www.journaldunet.com/web-tech/developpeur/1159810-bootstrap-definition-tutoriels-astuces-pratiques/) (Consulté le 10/08/2020).
- [22] [https : //www.supinfo.com/articles/single/7178 – demarrer – projet – avec – laravel](https://www.supinfo.com/articles/single/7178-demarrer-projet-avec-laravel) (Consulté le 10/08/2020).

## RÉSUMÉ

Le mémoire que nous avons présenté porte sur la conception et la réalisation d'une application web de gestion des annonces. Cette application web que nous avons nommé **SoukDzair** est facile à utiliser et permet de répondre aux besoins des annonceurs. Elle se traduit par la mise en œuvre d'un système qui facilite la gestion de tous types d'annonces.

Ce travail a été conçu en utilisant le processus de développement UP et le langage de modélisation UML. Pour ce qui est de la réalisation et de l'implémentation, notre choix s'est porté sur le Framework Laravel qui est basé sur les langages PHP, CSS, HTML, JavaScript et MYSQL pour la base de données, ces technologies qui nous ont offert un résultat appréciable et répondant mieux à nos attentes.

**Mots clés :** Laravel, PHP, UML, UP, application web, SoukDzair.

## ABSTRACT

The dissertation that we have presented concerns the design and production of a web application for managing advertisements. This web application that we have named **SoukDzair** is easy to use and responds to advertisers' needs. It results in the implementation of a system which facilitates the management of all types of advertisements.

This work was designed through using the UP development process and the UML modeling language. Regarding the realization and the implementation, our choice focused on the Framework Laravel which is based on the languages PHP, CSS, HTML, JavaScript and MYSQL for the database, these technologies offered us an appreciable result and better responding to our expectations.

**key words :** Laravel, PHP, UML, UP, web application, SoukDzair.