

Institut Supérieur de Comptabilités et D'administration des Entreprises

Nouakchott-Mauritanie



Rapport de stage

Filière : Informatique de Gestion

Stage effectuée du [01/08/2024] au [02/09/2024]

Nom de la société : SYSKAT Technologie



SYSKAT Technologie

Intitulé du Stage :

Développer une Site web :

« Gestion des Notaires »

Elaborer Par :
Med Mahfoudh

Encadrer par :
Aboubacar Bah

Année Universitaire : 2024 – 2025

Dédicace

Je dédie ce travail de fin d'étude à tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce projet.

À mes parents, pour leur amour, leur patience et leur soutien indéfectible tout au long de mon parcours académique. Sans eux, rien n'aurait été possible.

À mes enseignants et encadrants, pour leurs précieux conseils, leur disponibilité et leur accompagnement, qui ont enrichi mes connaissances et affiné mes compétences.

À mes amis et camarades de promotion, avec qui j'ai partagé des moments de travail, de doute, mais aussi de réussite et de satisfaction.

Enfin, à tous ceux qui aspirent à l'excellence et à l'innovation, en espérant que ce projet puisse servir d'inspiration et de motivation.

Merci à tous.

REMERCIEMENTS

Je n'aurai pas commencé ce rapport sans remercier ALLAH le tout puissant, le tout miséricordieux, qui m'a donné grâce et bénédiction pour faire ce stage.

Mes remerciements s'étendent également à Monsieur **Aboubekrine ould Bah**, Directeur Général de **SYSKAT Technologies**, pour m'avoir donné la chance d'intégrer **SYSKAT Technologie** et qui a su m'aider et me conseiller quand j'en avais besoin. Il est resté très à l'écoute et a su me faire confiance. Aussi mes enseignants durant ces années d'études auprès de qui mon beaucoup appris.

Mes remerciements s'adressent également aux membres de jury pour l'honneur qu'ils m'adressent en acceptant d'évaluer mon projet.

Ainsi je tiens à remercier les camarades de la dernière promotion qui m'ont permis d'en apprendre plus sur le fonctionnement d'une entreprise et les méthodes des recherches J'ai aussi remercié ma famille et surtout ma mère pour son constant soutien.

Résumé

Ce projet vise à optimiser la gestion des tâches administratives et juridiques d'un notaire en développant une application dédiée. L'objectif principal est d'automatiser et de centraliser les différentes opérations notariales, notamment la gestion des documents, le suivi des dossiers clients et la planification des rendez-vous.

La méthodologie adoptée repose sur une analyse approfondie des besoins des études notariales, suivie de la conception et du développement d'une solution numérique adaptée. L'application intègre des fonctionnalités de gestion documentaire, de signature électronique, de suivi des transactions et d'archivage sécurisé, tout en garantissant la conformité aux réglementations en vigueur.

Les résultats obtenus démontrent une amélioration significative de l'efficacité des processus, une réduction des délais de traitement et une meilleure organisation des tâches notariales. Cette solution contribue ainsi à renforcer la productivité des études notariales et à améliorer l'expérience client grâce à une gestion fluide et transparente.

Summary

This project aims to optimize the management of administrative and legal tasks for a notary by developing a dedicated application. The main objective is to automate and centralize various notarial operations, including document management, client file tracking, and appointment scheduling.

The adopted methodology is based on an in-depth analysis of the needs of notarial offices, followed by the design and development of a tailored digital solution. The application includes features for document management, electronic signatures, transaction tracking, and secure archiving, while ensuring compliance with current regulations.

The obtained results demonstrate a significant improvement in process efficiency, a reduction in processing times, and better organization of notarial tasks. This solution thus helps enhance the productivity of notarial offices and improve the client experience through smooth and transparent management.

الملخص:

يهدف هذا المشروع إلى تحسين إدارة المهام الإدارية والقانونية للمكاتب العدل من خلال تطوير تطبيق مخصص. الهدف الرئيسي هو أتمتة وتوحيد العمليات العدلية المختلفة، بما في ذلك إدارة الوثائق، متابعة ملفات العملاء، وتنظيم المواعيد.

تعتمد المنهجية المتبعة على تحليل دقيق لاحتياجات المكاتب العدلية، يليه تصميم وتطوير حل رقمي ملائم. يتضمن التطبيق ميزات لإدارة الوثائق، التوقيع الإلكتروني، تتبع المعاملات، والأرشفة الآمنة، مع ضمان الامتثال للأنظمة المعمول بها.

تُظهر النتائج المحققة تحسناً كبيراً في كفاءة العمليات، وتقليصاً في أوقات المعالجة، وتنظيماً أفضل للمهام العدلية. تساهم هذه الحلول في تعزيز إنتاجية المكاتب العدلية وتحسين تجربة العملاء من خلال إدارة سلسلة شفافة.

Table de matières

Liste des Acronymes.....	7
Table de figure :	8
INTRODUCTION GENERALE :	9
1. L'ENTREPRISE SYSKAT	10
1.Compétence de SYSKAT Technologie	11
2.Les solutions de SYSKAT	12
3.Organigramme de L'Organisation de SYSKAT	12
4.Présentation du problématique	13
2. ANALYSE ET SPÉCIFICATION DES BESOINS	14
1.INTRODUCTION :	15
2.Spécification des besoins :	15
1.Besoins fonctionnels :	15
2.Besoins non fonctionnels :	15
3.Analyse des besoins.....	16
1.Etude préalable :	16
2.Identification des acteurs :	16
4.Méthodologie de développement :	16
1.Définition de processus unifie :	16
2.Phase du processus Unifie :	16
3.Activités du processus unifie	17
5.Conclusion :	17
3. CONCEPTION	19
1.Présentation du langage UML	20
2.CONCEPTION DE L'APPLICATION.....	22
1.Modélisation avec le diagramme des cas d'utilisation :	22
2.DIAGRAMME DE CLASSE	24
3.Passage au modèle relationnelle	24
3.Conclusion	26
4. DEVELOPEMENT ET REALISATION.....	27
1.INTRODUCTION :	28
2.Architecture de l'application :	28
Model MVC :	28
Modèle :	28
3.Architecture du système :	30
1.Serveur de bases de données :	31

2.Environnement de Développement :.....	31
3.Environnements matériels :	31
4.Environnements logiciels :.....	32
5.Outils et logiciel.....	35
5. PRESENTATION DU SYSTEMES	40
1. INTRODUCTION :	41
2. Les Principales Interfaces graphique :	41
Page Principale :	41
Interfaces Notaire.....	42
3.Conclusion :	42
Conclusion Générale :	43
○ Bibliographie.....	44

Liste des Acronymes

Acronyme	Signification
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JavaScript	Langage de programmation permettant d'ajouter de l'interactivité aux pages web
UML	Unified Modeling Language (Langage de Modélisation Unifié)
BootStrap	Framework CSS pour le développement de sites web responsifs
Booch OMT	Booch Object Modeling Technique (Méthode de modélisation orientée objet développée par Grady Booch)
OMG	Object Management Group (Consortium qui développe les standards UML, CORBA, etc.)
Merise	Méthode de conception et de modélisation des systèmes d'information UML : Unified Modeling Language
OOSE	Object-Oriented Software Engineering (Ingénierie logicielle orientée objet)
THM	Three Horizons Model (Modèle des trois horizons pour la gestion de l'innovation)
MVC	Model-View-Controller (Modèle-Vue-Contrôleur, un modèle architectural en développement logiciel)
HTTP	HyperText Transfer Protocol (Protocole de transfert hypertexte)

Table de figure :

Figure 1: UML	20
Figure 2: Diagramme UML	21
Figure 3: Diagramme de cas d'utilisation.....	23
Figure 4: Diagramme de classe.....	24
Figure 5: Model MVS	28
Figure 6: Serveur web	30
Figure 7: Serveur de base de donnees	31
Figure 8: Machine HP	32
Figure 9: Logo HTML.....	32
Figure 10: Logo CSS.....	33
Figure 11: Logo javascript	33
Figure 12: Logo SQL	34
Figure 13: Logo PHP	34
Figure 14: Logo MYSQL	35
Figure 15: UML	36
Figure 16: Logo Bootstrap	38
Figure 17: Visual Studio Code	38
Figure 18: Logo Laravel.....	39
Figure 19: Interface de saisie	41
Figure 20: Interface de Notaire	42
Figure 21: Archive.php.....	45
Figure 22: Search.php	45
Figure 23: Search.php	46
Figure 24:Document.php.....	46

INTRODUCTION GENERALE :

La numérisation des bureaux de notaires constitue un défi évident pour améliorer l'organisation et la gestion des affaires juridiques. Depuis plus de vingt ans, la technologie informatique a pénétré les bureaux de notaires. De nos jours, ces avancées font l'objet de projets de numérisation visant à améliorer la gestion des affaires juridiques. Avant l'invention de l'ordinateur, toutes les informations étaient enregistrées manuellement sur des supports papier, ce qui entraînait de nombreux problèmes tels que la perte de temps considérable lors de la saisie et de la recherche, leur dégradation due à une consultation fréquente et leur sécurisation, etc.

Ce même problème persiste même dans les bureaux de notaires, ce qui en fait une partie intégrante des structures que la technologie informatique peut grandement aider. C'est dans cette optique que nous nous sommes intéressés aux bureaux de notaires. Ce choix est motivé par le besoin d'améliorer le système de gestion des affaires et les problèmes liés à la gestion des données. Ce dernier a fait que chaque département a son propre système de numérotation, ce qui ne permet pas un bon suivi de toutes les affaires. Par conséquent, on note une redondance récurrente et une perte de temps dans la saisie et la recherche des informations des affaires.

Pour une bonne structuration, j'ai organisé mon travail en cinq chapitres. Dans le premier chapitre, je vais présenter le cadre général de mon stage. Dans les deuxième et troisième chapitres, je vais analyser et spécifier les besoins exprimés pour passer à la conception. Ensuite, je suis passé à la réalisation dans le quatrième chapitre pour terminer avec une présentation globale du système dans le cinquième chapitre

1. L'ENTREPRISE SYSKAT

INTRODUCTION

SYSKAT Technologies **SARL** est une société Mauritanienne. Elle serve les particuliers, les petites, moyennes et grandes entreprises, cette entreprise née 2007. Elle est spécialisée dans le domaine de la conception, création et administration des bases de données, Développement des applications, Traitement des données.

Elle propose une offre de services dans le conseil en systèmes d'informations, le développement de logiciels, l'intégration, la conduite de projets et le transfert de compétences. Comme toute Société de services en ingénierie informatique(**SSI**) qui vise la performance, **SYSKAT** est à l'écoute du marché et cherche toujours à avoir une satisfaction totale du client.

1.Compétence de SYSKAT Technologie



✚ Ingénierie

- Conception, création et administration des bases de données
- Développement des applications
- Traitement des données

✚ Web



- ✓ Stratégie et ergonomie
- ✓ Rédaction optimisée
- ✓ Conception
- ✓ Création
- ✓ Analyse des résultats
- ✓ Analyse des résultats

✚ Sécurité et administration des réseaux



DMZ

Supervisions de réseaux et de
Systèmes d'exploitation

Mise en place de mécanismes
d'authentification SSO

- Mise en place des réseaux internes Ethernet et WIFI

✚ Technologies



.NET, PHP.

- Langages : JAVA, C, C++, SQL, PL/SQL,

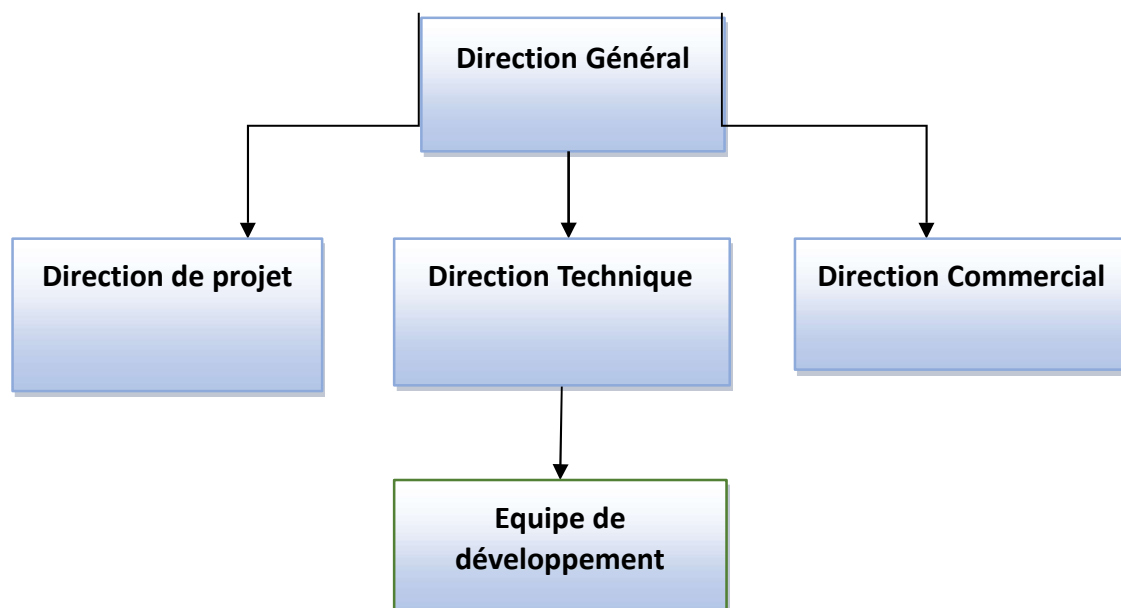
- Platform : Linux, Windows
- Frameworks: JSF, STRUTS, Hibernate, spring, Design Pattern, Maven, Ant, Zend.

2.Les solutions de SYSKAT

SYSKAT a des solutions développées à partir des technologies **JAVA/JEE** avec l'utilisation des Framework modernes tels que **JSF**, Hibernante, **Spring**...

- **Système d'Information Hospitalier Intégré « SIH »**
- **Gestion Complète d'une Clinique « ESSIHA »**
- **Gestion des Ressources Humaines « GRH »**
- **Gestion de comptabilité générale « COMPTA »**
- **Gestion commerciale « GESCOM »** • **Centre messagerie « AL-MERSAL »**
- etc.

3.Organigramme de L'Organisation de SYSKAT



4.Présentation du problématique

Une bonne gestion de programme opératoire demeure nécessaire et indispensable pour un bon suivi et une bonne prise en charge des malades. Même s'il y'en a dans la plupart des établissements hospitaliers, on a pu constater que ces derniers optent généralement pour les méthodes classiques, c'est-à-dire toutes les informations du patient pour une opération sont enregistrées manuellement. Cette méthode présente beaucoup de limites telles qu'une perte de temps considérable dans la saisie et la recherche de l'information du patient. C'est dans ce cadre que nous avons jugé nécessaire de mettre en place une application pour la gestion des programme opératoires afin d'améliorer cette dernière.

CONCLUSION

SYSKAT Technologie est une entreprise spécialisée dans le développement de solutions informatiques adaptées aux besoins des entreprises et des professionnels. Grâce à son expertise dans le domaine des technologies numériques, elle propose des services innovants, notamment en matière de développement web et de gestion des données

2. ANALYSE ET SPÉCIFICATION DES BESOINS

1.INTRODUCTION :

Le travail qui m'a été confié pendant ma période de stage est de développer une application web visant à documenter les processus d'achat et de vente. Pour cela, il est nécessaire de mener une étude sur ce qui existe déjà sur le marché afin de comprendre tout d'abord comment fonctionnent les applications actuelles, et deuxièmement, les points sur lesquels je vais travailler. Cependant, en conséquence, il n'y a pas d'application web de ce type sur le marché, donc les exigences fonctionnelles et non fonctionnelles de l'application ont été définies non pas par une étude de l'existant, mais par des échanges de points de vue entre moi et mon superviseur.

2.Spécification des besoins :

Dans cette partie, on explique en détail ce que l'application est censée faire et ceci à travers la spécification des besoins fonctionnels et non fonctionnels.

1.Besoins fonctionnels :

Les besoins fonctionnels ou besoins métiers représentent les actions que le système doit exécuter.

Cette application doit satisfaire principalement le besoin fonctionnel suivant :

○Gestion de Notaire

2.Besoins non fonctionnels :

Ce sont des exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt ils identifient des contraintes internes et externes du système. Les principaux besoins non fonctionnels de mon application se résument dans les points suivants :

- **Performance**
 - L'application répond à tous les exigences de l'utilisateur d'une manière optimale
- **Fiabilité**○Bon fonctionnement de l'application sans détection de défaillance
- **Rapidité**
 - Le déplacement entre les pages sera facile et rapide
- **Convivialité**
 - Un design clair, souple et interactif
 - Une bonne interface qui donne l'envie à l'utilisateur d'utiliser l'application

- Positionnement du contenu dans les pages de la manière la plus accessible
- **Portabilité**○L'application est multiplateforme : Elle fonctionne sur tout système d'exploitation

3.Analyse des besoins

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.

1.Etude préalable :

L'étude préalable est une partie primordiale dans la réalisation d'un projet informatique. Pour atteindre nos buts et objectifs, il est nécessaire d'avoir une vue claire des différents besoins escompter. Raison pour laquelle on est appelé à faire une étude des méthodes ou systèmes qu'utilise l'hôpital pour la gestion des programmes opératoires.

2.Identification des acteurs :

Un acteur représente l'abstraction d'un rôle joué par des entités externe (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. Dans mon application le Chirurgienne et Assistante sont les seuls acteurs qui interagissent avec le système.

4.Méthodologie de développement :

Contenu de la nature de mon projet, j'ai jugés que la méthode PU (processus unifié) serait la plus adapté pour sa réalisation.

1.Définition de processus unifie :

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.

2.Phase du processus Unifie :

La méthode UP se base sur quatre phases :

- **Analyse des besoins** : Établir une vision globale du projet où on spécifie Les besoins et on étudie la faisabilité du projet.
- **Élaboration** : On reprend les éléments de l'analyse des besoins et on développe une architecture de référence, les risques et la plupart des besoins sont identifiés.
- **Construction** : Finaliser l'analyse, la conception, l'implémentation et les tests puis transformer l'architecture de référence en produit exécutable tout en veillant à respecter son intégrité.
- **Transition** : Livraison du produit au client afin d'effectuer des essais pour détecter d'éventuelles anomalies.

3. Activités du processus unifié

Chaque phase est constituée d'une succession d'activités. Les activités de la méthode UP sont :

- **Expression des besoins** : Compréhension et expression des besoins et des exigences du client qu'elles soient fonctionnelles ou non fonctionnelles.
- **Analyse et Conception** : Permet d'acquérir une compréhension approfondie des contraintes liées aux outils de réalisation en prenant en compte le choix d'architecture technique retenu pour le développement et l'exploitation système.
- **Implémentation** : On implémente le système sous forme de composants, bibliothèques et de fichiers. Elle a pour objectif de planifier l'intégration.
- **Tests** : Permettent de vérifier les résultats de l'implémentation de toutes les exigences et de s'assurer de la bonne intégration de tous les composants dans le logiciel. ○ **Déploiement** : Livraison et exploitation du produit

5. Conclusion :

Ce chapitre a été dédié à la spécification des besoins après une étude détaillée de l'application : son objectif et son fonctionnement.

J'ai consacré la deuxième semaine de mon stage à cette étude ce qui m'a donné une bonne conscience de ses besoins fonctionnels et non fonctionnels.

Cette étude m'a permis de distinguer les différents points sur lesquels je vais travailler et les technologies que je devrais utiliser pour réaliser mon application.

3. CONCEPTION

INTRODUCTION

La démarche de conception est une étape fondamentale dans le processus de développement puisqu'elle fait correspondre la vision applicative (le modèle d'analyse) à la vision technique (l'environnement de développement et d'exécution).

Ce chapitre vise à illustrer la phase de conception et les modèles **UML** associés. J'ai commencé par établir les diagrammes des cas d'utilisation, ensuite et le diagramme de classe, et je termine avec une petite conclusion.

1.Présentation du langage UML



Figure 1: UML

Pour faire face à la complexité des systèmes d'information, de nouvelles méthodes et outils ont été créées. La principale avancée des quinze dernières années réside dans la programmation orientée objet.

Face à ce nouveau mode de programmation, les méthodes de modélisation classique (telle que **MERISE**) ont rapidement montré certaines limites et ont dû s'adapter.

De très nombreuses méthodes de modélisation ont également vu le jour comme **Boch**, **OMT**

...

Dans ce contexte et devant le foisonnement de nouvelles méthodes de conception orientée objet, l'**OMG** (Object Management Group) a eu comme objectif de définir une notation standard utilisables dans les développements informatiques basés sur l'objet. C'est ainsi qu'est apparu **UML** (qui signifie en français langage de modélisation unifiée), qui est issu de la fusion des méthodes de **Booch**, **OMT** et **OOSE**. C'est un langage de modélisation qui permet de représenter graphiquement les besoins des utilisateurs à l'aide de diagramme.

Un diagramme **UML** est une représentation graphique, qui s'intéresse à un aspect précis du modèle. C'est une perspective du modèle, pas « le modèle ». Chaque type de diagramme **UML** possède une structure.

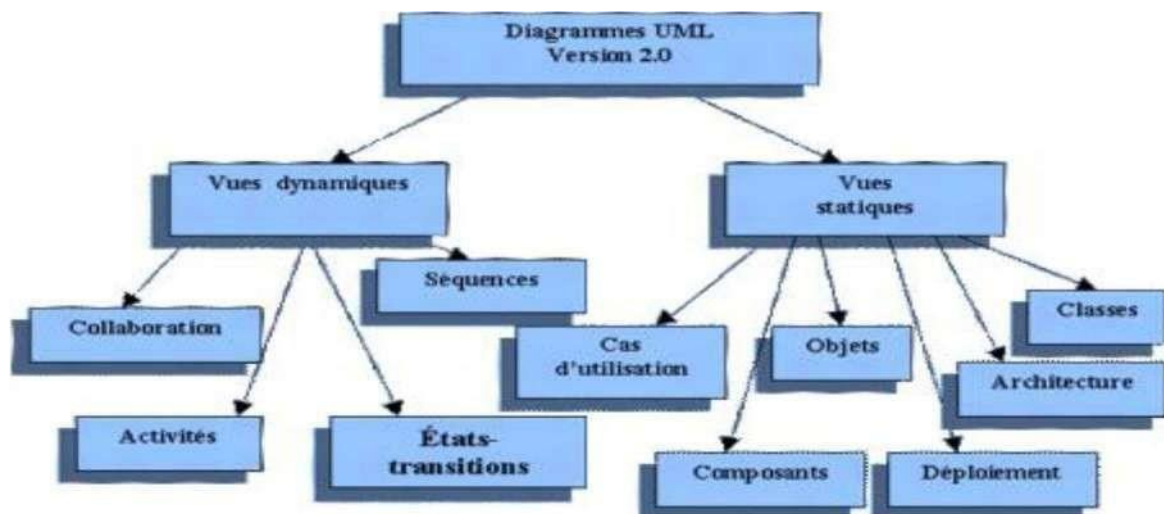


Figure 2: Diagramme UML

Il existe deux types de vues du système constituant chacune des diagrammes qui sont répartis selon leurs aspects statiques ou dynamiques.

Selon les vues statiques nous avons :

- Le diagramme des cas d'utilisation

Le diagramme de cas d'utilisation permet de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. Il essaiera de répondre à des questions du genre : Qui devra pouvoir faire quoi grâce au logiciel. Les acteurs principaux qui sont liés à un package auront besoin de cette partie du logiciel pour réaliser une plusieurs lots d'actions.

- Le diagramme d'objet

Le diagramme d'objet sert à illustrer les classes complexes en utilisant des exemples d'instances. A l'exception de la multiplicité, qui est explicitement indiquée, le diagramme d'objets utilise les mêmes concepts que le diagramme de classes. Ils sont essentiellement utilisés pour comprendre ou illustrer des parties complexes d'un diagramme de classes.

- Le diagramme des classes

Le diagramme des classes représente les entités manipulées par les utilisateurs. L'intérêt du diagramme des classes est de modéliser les entités du système d'information. Le diagramme met en évidence d'éventuelles relations entre les classes ou les entités.

- Le diagramme de déploiement

Les diagrammes de déploiement correspondent à la description de l'environnement d'exécution du système (matériel, réseau...) et de la façon dont les composants y sont installés. Les diagrammes de déploiement sont donc très utiles pour modéliser l'architecture physique d'un système.

Selon les vues dynamiques nous avons :

- Le diagramme de collaboration

Le diagramme de collaboration (appelé également diagramme de communication) permet de mettre en évidence les échanges de messages entre objets. Cela nous aide à voir clair dans les actions qui sont nécessaires pour produire ces échanges de messages. Et donc de compléter, si besoin, les diagrammes de séquence et de classes.

- Le diagramme de séquence

Le diagramme de séquence permet de décrire les différents scénarios d'utilisation du système. C'est une variante du diagramme de collaboration sauf qu'il possède intrinsèquement une dimension temporelle mais ne représente pas explicitement les liens entre les objets.

- Le diagramme d'états-transitions

Le diagramme d'état-transition permet de décrire le cycle de vie des objets d'une classe. Il définit l'enchaînement des états de classe et font donc apparaître l'ordonnancement des travaux.

- Le diagramme d'activités

Le diagramme d'activité représente le déroulement des actions, sans utiliser les objets. En phase d'analyse, il est utilisé pour consolider les spécifications d'un cas d'utilisation.

Conscient de la complémentarité qui existe entre ces différents diagrammes, et sachant qu'UML ne préconise aucune démarche, on a jugé nécessaire de ne prendre que certains de ces diagrammes qui répondent bien à nos besoins afin de concevoir notre système.

Cependant, les diagrammes qui vont être utilisés dans la suite de mon exposé sont les suivants :

Le diagramme des cas d'utilisation ;

Et Le diagramme des classes.

2.CONCEPTION DE L'APPLICATION

1.Modélisation avec le diagramme des cas d'utilisation :

a. Diagramme des cas d'utilisation

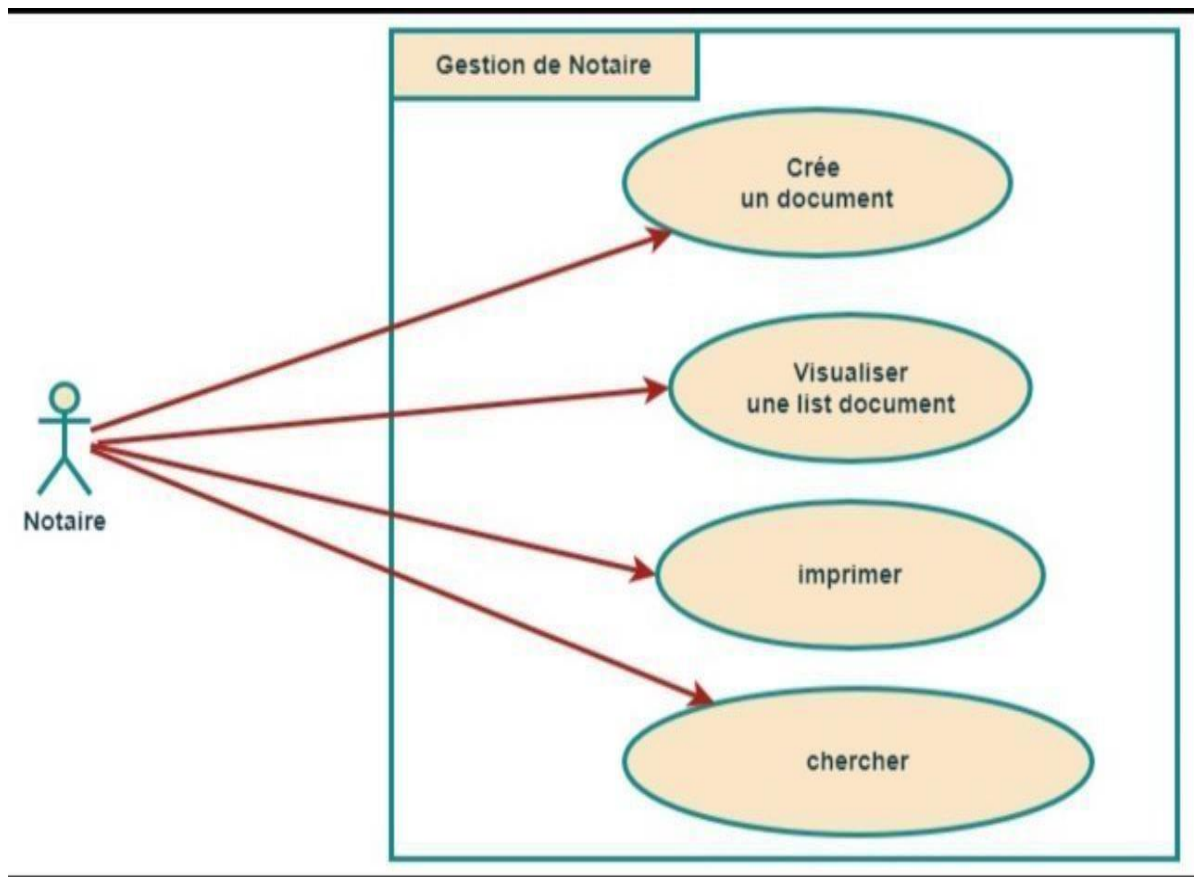


Figure 3: Diagramme de cas d'utilisation

2.DIAGRAMME DE CLASSE

Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre eux; Son intérêt est de modéliser les entités du système d'information. Le diagramme de classe permet de représenter l'ensemble des informations finalisées qui sont gérées par le domaine. Ces informations sont structurées, c'est-à-dire qu'elles sont regroupées dans des classes.

Le diagramme met en évidence d'éventuelles relations entre ces classes. Le diagramme de classes de mon application est le suivant

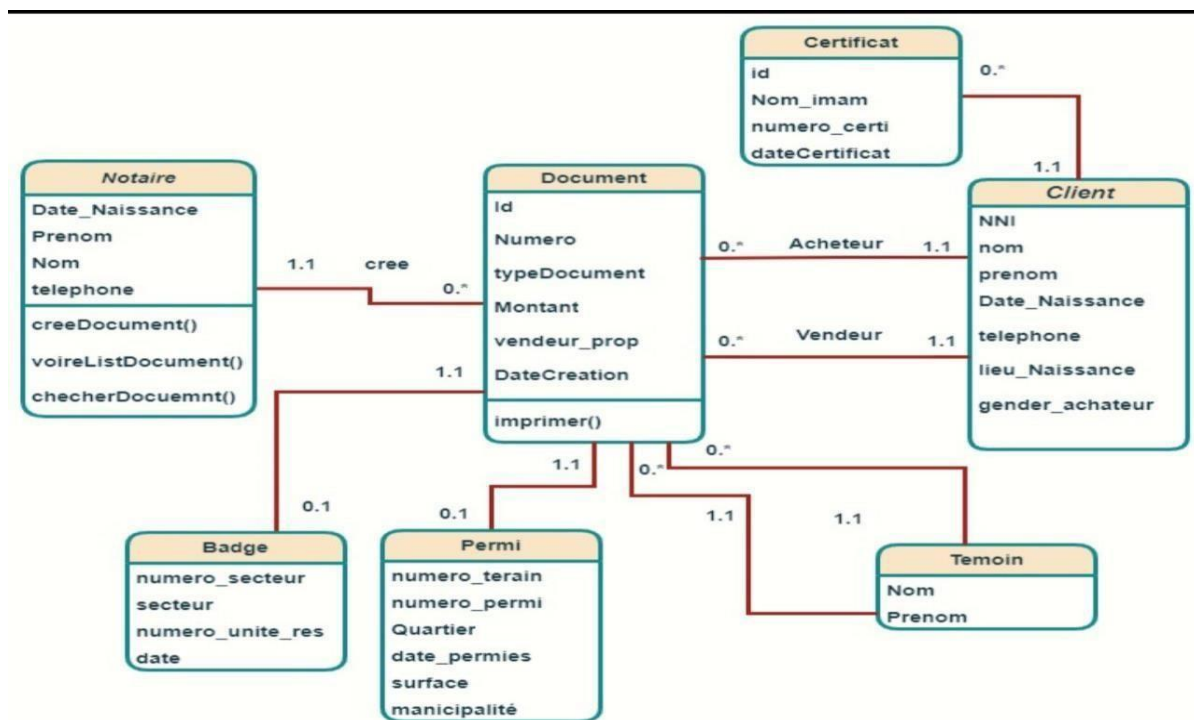


Figure 4: Diagramme de classe

3.Passage au modèle relationnelle

Le modèle relationnel est le modèle logique de donnée qui correspond à l'organisation des données dans les bases de données relationnelles.

Un modèle relationnel est composé de relations, encore appelée table. Ces tables sont décrites par des attributs aux champs. Pour décrire une relation, on indique tout simplement son nom, suivi du nom de ses attributs entre parenthèses.

L'identifiant d'une relation est composé d'un ou plusieurs attributs qui forment la clé primaire. Une relation peut faire référence à une autre en utilisant une clé étrangère, qui correspond à la clé primaire de la relation référencée.

Le modèle logique de mon application est le suivant :

3.Conclusion

Dans ce chapitre, j'ai présenté mon étude conceptuelle du système. La vue dynamique nous a permis d'avoir une vue générale sur le déroulement des cas d'utilisation et leurs exécutions, cette vue a été modélisé par des diagrammes de séquence du système puis par des diagrammes d'activités.

La vue statique, réalisée par le diagramme des classes nous a permis de définir la structure du système et de dégager les différentes entités y composés.

Enfin la conception graphique nous a permis dereprésenter les différentes sortes de l'application.

4. DEVELOPEMENT ET REALISATION

1.INTRODUCTION :

Après l'étape de conception de l'application, nous allons dans ce chapitre, d'écrire la phase de réalisation et de développement. Nous allons présenter, en premiers lieu, L'architecture de l'application et du Système ensuite parlerons de l'environnement du travail utilisé pour le développement de l'application.

2.Architecture de l'application :

Model MVC :

Le développement de mon application s'est basé sur l'architecture **MVC**. Ce paradigme divise l'**IHM** en un modèle, une vue et un contrôleur, chacun ayant un rôle bien précis.

Cette architecture permet de bien organiser son code source en nous aidant à savoir quels fichiers créer, mais surtout à définir leur rôle.

Ce maître d'architecture impose la séparation entre les données, la présentation et les traitements, ce qui nous donne trois parties fondamentales dans l'application : le modèle, la vue et le contrôleur.



Figure 5: Model MVS

Modèle :

Cette partie illustre le comportement de notre application : traitement des données et interaction avec la base de données. 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. Le modèle offre des méthodes pour mettre à jour les données (ajout, suppression, modification). On y trouve donc entre autres les requêtes **SQL**.

Vue :

Cette partie se concentre sur l'affichage et correspond à l'interface avec laquelle l'utilisateur interagit. 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).

être traitées par le contrôleur. Le modèle offre des méthodes pour mettre à jour les données (ajout, suppression, modification). On y trouve donc entre autres les requêtes **SQL**.

Vue :

Cette partie se concentre sur l'affichage et correspond à l'interface avec laquelle l'utilisateur interagit. 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).

3. Architecture du système :

L'architecture client-serveur s'appuie sur un poste central, le serveur, qui envoie des données aux machines clientes.

Serveur web :

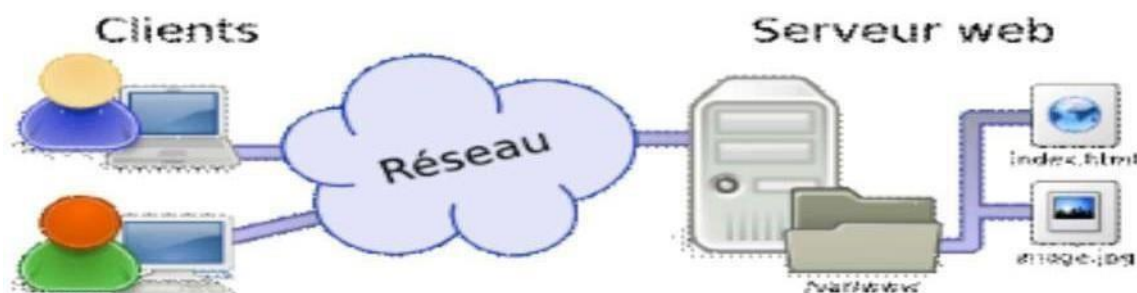


Figure 6: Serveur web

Un serveur web peut faire référence à des composants logiciels (**software**) ou à des composants matériels (**hardware**) ou à des composants logiciels et matériels qui fonctionnent ensemble. Au niveau des composants matériels, un serveur web est un ordinateur qui stocke les fichiers qui composent un site web (par exemple les documents **HTML**, les images, les feuilles de style **CSS**, les fichiers **JavaScript**) et qui les envoie à l'appareil de l'utilisateur qui

visite le site. Cet ordinateur est connecté à Internet et est généralement accessible via un nom de domaine tel que **mozilla.org**. (Introduction, 2024)

Au niveau des composants logiciels, un serveur web contient différents fragments qui contrôlent la façon dont les utilisateurs peuvent accéder aux fichiers hébergés. On trouvera à minima un serveur **HTTP**. Un serveur **HTTP** est un logiciel qui comprend les **URL** et le protocole **HTTP**.

1. Serveur de bases de données :



Figure 7: Serveur de base de données

Un serveur de base de données sert à stocker, à extraire et à gérer les données dans une base de données. Il permet également de gérer la mise à jour des données. Il donne un accès simultané à cette base à plusieurs serveurs web et utilisateurs. Enfin, il assure la sécurité et l'intégrité des données. Et quand on parle de données, on entend peut-être des millions d'éléments simultanément accessibles à des milliers d'utilisateurs.

En plus de ces fonctions principales, le logiciel de serveur de base de données offre des outils qui facilitent et accélèrent l'administration de la base, comme l'exportation de données, la configuration de l'accès de l'utilisateur et la sauvegarde des données. (Gestion Bibliotheque, 2024)

2. Environnement de Développement :

Pour interagir avec le serveur et la base de données, nous sommes appelés à faire recours au moins à un langage de programmation. Dans cette partie j'ai identifié les différentes caractéristiques de l'environnement matériel et logiciel que j'aiservi à l'implémentation de mon application.

3. Environnements matériels :

La machine utilisée pour réaliser ce projet :

Ordinateur portable



Figure 8: Machine HP

Machine HP

Mémoire Vive : 8 Go Disque

Dur : 300 Go

Processeur : Intel (R)

Type de système : Windows 10

4. Environnements logiciels :

Les langages de programmation

- **HTML**



Figure 9: Logo HTML

L'HyperText Mark up Langage, généralement abrégé **HTML**, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom. **HTML** permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques.

Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec le

langage de programmation JavaScript et des feuilles de style en cascade (**CSS**). **HTML** est inspiré du Standard Generalized Markup Language (**SGML**). Il s'agit d'un format ouvert. (Gestion Bibliotheque, 2024)

- **CSS**



Figure 10: Logo CSS

Les feuilles de style en cascade¹, généralement appelées **CSS** de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents **HTML** et **XML**. Les standards définissant **CSS** sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, **CSS** devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000. (CSS Documentation, 2024)

- **Java Script**



Figure 11: Logo javascript

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs avec l'utilisation (par exemple) de **Node.js**. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe. Le langage supporte le paradigme objet, impératif et fonctionnel. **JavaScript** est le langage possédant le plus large écosystème grâce à son gestionnaire de dépendances **NPM**, avec environ 500 000 paquets en août 2017.

JavaScript a été créé en 1995 par Brendan Eich. Il a été standardisé sous le nom **d'ECMAScript** en juin 1997 par **Ecma** International dans le standard **ECMA-262**. Le standard **ECMA-262** en est actuellement à sa 8e édition. JavaScript n'est depuis qu'une implémentation **d'ECMAScript**, celle mise en œuvre par la fondation Mozilla. L'implémentation **d'ECMAScript** par Microsoft (dans Internet Explorer jusqu'à sa version 9) se nomme **JScript**, tandis que celle d'Adobe Systèmes se nomme Action Script. (Gestion Bibliothèque, 2024)

Avec les technologies **HTML** et **CSS**, **JavaScript** est parfois considéré comme l'une des technologies cœur du World Wide Web. Le langage **JavaScript** permet des pages web interactives, et à ce titre est une partie essentielle des applications web. Une grande majorité des sites web l'utilisent, et la majorité des disposent d'un moteur **JavaScript** dédié pour l'interpréter, indépendamment des considérations de sécurité qui peuvent se poser dans le cas échéant. (HTML Introduction, 2024)

- **SQL**



Figure 12: Logo SQL

SQL est un langage informatique utilisé pour effectuer des requêtes sur des bases de données ou systèmes d'information. Il permet d'obtenir les données vérifiant certaines conditions (on parle de critères de sélection). Les données peuvent être triées, elles peuvent également être regroupées suivant les valeurs d'une donnée particulière. **SQL** est le langage de requête le plus connu et le plus utilisé. (SQL Docs, 2024)

- **PHP**



Figure 13: Logo PHP

PHP est un langage informatique utilisé sur l'internet. Le terme PHP est un acronyme récursif de « **PHP : HyperTextPreprocessor** ».

Ce langage est principalement utilisé pour produire un site web dynamique. Il est courant que ce langage soit associé à une base de données, tel que **MySQL**. Exécuté du côté serveur (l'endroit où est hébergé le site) il n'y a pas besoin aux visiteurs d'avoir des logiciels ou plugins particulier. Néanmoins, les webmasters qui souhaitent développer un site en **PHP** doivent s'assurer que l'hébergeur prend en compte ce langage.

Lorsqu'une page **PHP** est exécutée par le serveur, alors celui-ci renvoie généralement au client (aux visiteurs du site) une page web qui peut contenir du **HTML**, **XHTML**, **CSS**, **JavaScript...** (PHP Docs, 2024)

Argumentaire du choix de PHP

Tout d'abord, **PHP** est gratuit et ne nécessite aucune licence d'utilisation. Ensuite, **PHP** est le langage de programmation Web le plus utilisé au monde. Il existe une communauté de développeurs très active qui rend disponibles des dizaines de milliers de librairies **PHP** de grande qualité ainsi qu'une vaste quantité de documentation et tutoriels accessible à tous au bénéfice de chacun. Ces ressources facilitent notre travail et réduisent notre temps de programmation ce qui se traduit aussi en économie pour le client.

En termes de rapidité et d'efficacité, **PHP** n'a rien à envier aux autres langages. Plusieurs portails très populaires et nécessitant beaucoup de performance l'utilisent.

Nous avons qu'à penser à Facebook, Wikipédia, le réseau **CBC**, l'université Harvard, pour ne nommer que ceux-là. D'ailleurs le populaire système de gestion de contenu **WordPress** est lui-même construit en **PHP**. Ce dernier représente à lui seul environ 80% des sites Internet sur la toile. Une si grande popularité n'est certainement pas une coïncidence !

5. Outils et logiciel

- **MySQL**



Figure 14: Logo MYSQL

MySQL est un Système de Gestion de Base de Données (**SGBD**) parmi les plus populaires au monde. Il est distribué sous double licence, une licence publique générale **GNU** et une

propriétaire selon l'utilisation qui en est faite. La première version de MySQL est apparue en 1995 et l'outil est régulièrement entretenu.

Ce système est particulièrement connu des développeurs pour faire partie des célèbres quatuors:

WAMP (Windows, Apache, **MySQL** et **PHP**), **LAMP** (Linux) et **MAMP** (Mac). Ces packages sont si populaires et simples à mettre en œuvre que **MySQL** est largement connu et exploité comme système de gestion de base de données pour des applications utilisant PHP.

C'est d'ailleurs pour cette raison que la plupart des hébergeurs web proposent PHP et MySQL. (Gestion Bibliothèque, 2024)

Caractéristiques

MySQL est un serveur de base de données relationnelles SQL qui fonctionne sur de nombreux systèmes d'exploitation (dont Linux, Mac OS X, Windows, Solaris, FreeBSD...) et qui est accessible en écriture par de nombreux langages de programmation, incluant notamment PHP, Java, Ruby, C, C++, .NET, Python ...

L'une des spécificités de MySQL c'est qu'il inclut plusieurs moteurs de bases de données et qu'il est par ailleurs possible au sein d'une même base de définir un moteur différent pour les tables qui composent la base. Cette technique est astucieuse et permet de mieux optimiser les performances d'une application. Les 2 moteurs les plus connus étant MyISAM (moteur par défaut) et InnoDB.

La réplication est possible avec MySQL et permet ainsi de répartir la charge sur plusieurs machines, d'optimiser les performances ou d'effectuer facilement des sauvegardes des données. (Gestion Bibliothèque, 2024)

•UML



Figure 15: UML

UML ("Unified Modeling Language," ou "langage de modélisation objet unifié") est né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 90 : **OMT**, **Booch** et **OOSE**.

Issu "du terrain" et fruit d'un travail d'experts reconnus, **UML** est le résultat d'un large consensus. De très nombreux acteurs industriels de renom ont adopté **UML** et participent à son développement.

En l'espace d'une poignée d'années seulement, **UML** est devenu un standard incontournable. La presse spécialisée foisonne d'articles exaltés et à en croire certains, utiliser les technologies objet sans **UML** relève de l'hérésie. Lorsqu'on possède un esprit un tant soit peu critique, on est en droit de s'interroger sur les raisons qui expliquent un engouement si soudain et massif ! **UML** est-il révolutionnaire ? L'approche objet est pourtant loin d'être une idée récente. Simula, premier langage de programmation à implémenter le concept de type abstrait à l'aide de classes, date de 1967 ! En 1976 déjà, **Smalltalk** implémente les concepts fondateurs de l'approche objet : encapsulation, agrégation, héritage. Les premiers compilateurs **C++** datent du début des années 80 et de nombreux langages orientés objets "académiques" ont étayés les concepts objets (Eiffel, Objective C, **Loops...**). (Gestion Bibliotheque, 2024)

Il y a donc déjà longtemps que l'approche objet est devenue une réalité. Les concepts de base de l'approche objet sont stables et largement éprouvés. De nos jours, programmer "objet", c'est bénéficier d'une panoplie d'outils et de langages performants. L'approche objet est une solution technologique incontournable.



Figure 16: Logo Bootstrap

Bootstrap est un Framework gratuit et à code source ouvert destiné au développement **Web** frontal réactif et premier mobile. Il contient des modèles de conception basés sur **CSS** et (éventuellement) JavaScript pour la typographie, les formulaires, les boutons, la navigation et d'autres composants d'interface.

Bootstrap est le troisième projet le plus étoilé sur **Git Hub**, avec plus de 131000 étoiles, derrière seulement **freeCodeCamp** (près de 300 000 étoiles) et légèrement derrière le cadre de **Vue.js**. (Boot Docs, 2024)

- **Visual Studio Code**



Figure 17: Visual Studio Code

Visual Studio Code est un éditeur de code multiplateforme édité par Microsoft. Cet outil destiné aux développeurs supporte plusieurs dizaines de langages de programmation comme le **HTML**, **C++**, **PHP**, **JavaScript**, **Markdown**, **CSS**, etc.

Visual Studio Code intègre plusieurs outils facilitant la saisie de code par les développeurs comme la coloration syntaxique ou encore le système d'auto-complétion

IntelliSense. En outre, l'outil permet aux développeurs de corriger leur code et de gérer

Les différentes versions de leurs fichiers de travail puisqu'un module de débogage est aussi de la partie. (VS Docs, 2024)

- **Laravel**



Figure 18: Logo Laravel

Créé en 2011 par *Taylor Otwell*, Laravel est devenu le Framework **PHP** libre et open source le plus populaire au monde. Il est utilisé dans le développement d'application web tout en suivant l'architecture **MVC** et basé sur **Symfony**. Comme chaque version de laravel est documentée de manière détaillée, on peut l'apprendre facilement et compter sur une mise à jour continue de sa base de connaissances.

Laravel est conçu pour le développement rapide d'applications. L'infrastructure dispose d'un moteur de Template bien construit qui permet une grande variété de tâches courantes telles .(Laravel, 2024)

CONCLUSION

La syntaxe raffinée, la possibilité d'exécuter des tâches en arrière-plan de manière asynchrone pour améliorer les performances et l'intégration bien établie avec Amazon Web (**AWS**) s'ajoute à la liste des fonctionnalités qui font de laravel l'un des meilleurs frameworks PHP Pour la réalisation de mon projet je l'ai choisi pour plusieurs raisons :

Laravel convient au développement d'application avec des besoins complexes, qu'ils soient petits ou grands.

C'est un Framework **PHP** complet avec des fonctionnalités qui nous aideront à personnaliserdes applications complexes.

Laravel est très expressif. Sa vitesse et sa sécurité répondent aux attentes d'une application web moderne. Laravel est la voie à suivre pour les développeurs qui souhaitent créer des applications d'entreprise qui évolueront en fonction des tendances du **web**. (Laravel, 2024)

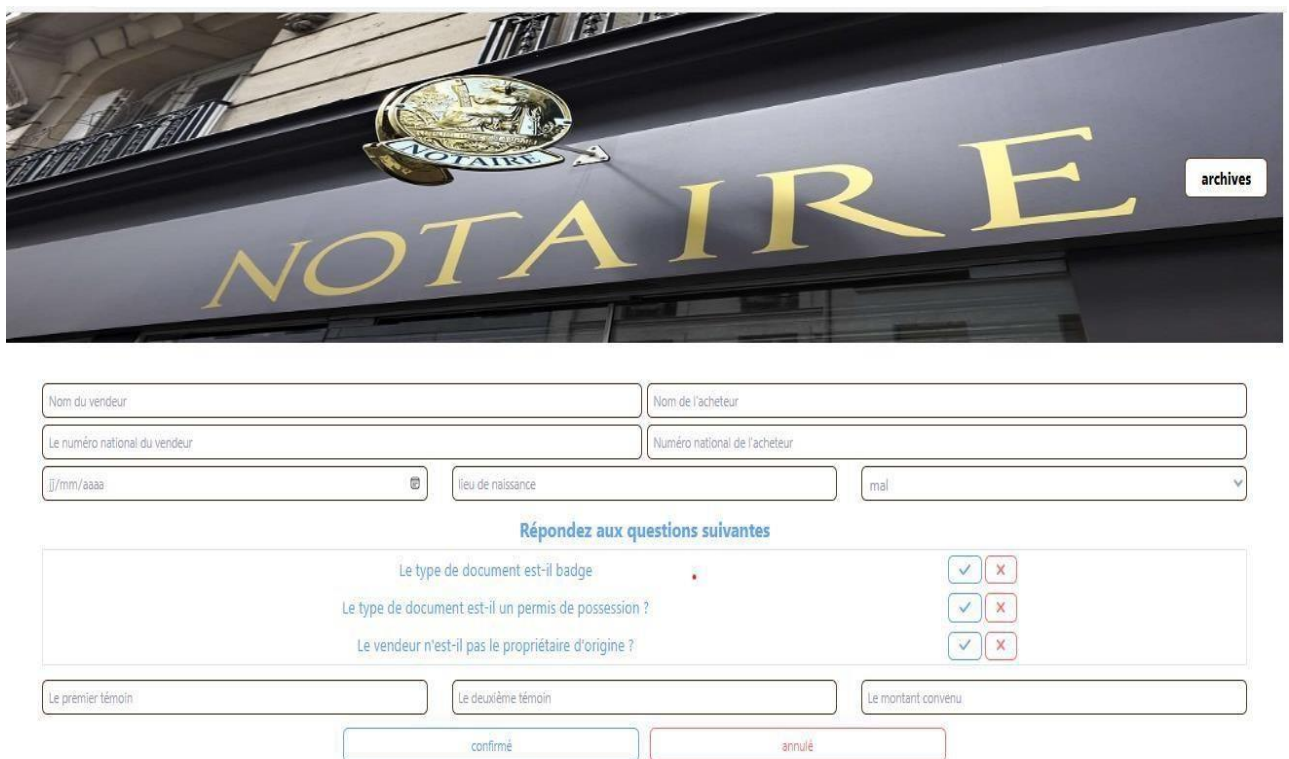
5. PRESENTATION DU SYSTEMES

1. INTRODUCTION :

Au cours de ce chapitre j'essaierai de faire une présentation générale de l'application. Cette présentation s'articulera sur les principaux types d'interfaces utilisateurs proposés par mon système. Rappelons qu'une interface utilisateur est une partie spécifique de l'application destinée à un utilisateur. Cette partie lui permettra d'interagir avec le système et de profiter aux différentes fonctionnalités qui lui sont offertes.

2. Les Principales Interfaces graphique :

Page Principale :



The screenshot displays the main interface of the 'NOTAIRE' application. At the top, there is a header featuring a circular logo with a notary seal and the word 'NOTAIRE' in large, gold, serif capital letters. To the right of the header is a small 'archives' button. Below the header is a registration form with the following fields:

- Nom du vendeur
- Nom de l'acheteur
- Le numéro national du vendeur
- Numéro national de l'acheteur
- jj/mm/aaaa (date format)
- lieu de naissance
- mail (dropdown menu)

Below these fields is a section titled 'Répondez aux questions suivantes' (Answer the following questions). It contains three questions, each with a '✓' (confirm) and an '✗' (cancel) button:

- Le type de document est-il badge
- Le type de document est-il un permis de possession ?
- Le vendeur n'est-il pas le propriétaire d'origine ?

At the bottom of the form are three input fields:

- Le premier témoin
- Le deuxième témoin
- Le montant convenu

Below these fields are two buttons: 'confirmé' (confirm) and 'annulé' (cancel).

Figure 19: Interface de saisie

À l'entrée du site, les champs suivants apparaissent :

- Nom du vendeur
- Nom du client

- Numéro national du vendeur
- Numéro national du client
- Date du contrat
- Lieu de naissance
- Sexe
- Nom du premier témoin
- Nom du deuxième témoin -Montant convenu

En plus des champs supplémentaires selon les questions posées.

Interfaces Notaire

The screenshot displays a web-based notary interface. At the top, there is a search bar. Below it, the header includes the name of the notary, 'Tevragh Zeyna Mosque Saudiya Mauritel', and contact information. The main body of the document contains a date stamp 'date du : 2024-10-20 12:04:25' and a paragraph of text in French describing a property transaction. Below the text, there are five designated areas for signatures and stamps: 'Signature et cachet de l'imam' (with a circular notary seal), 'Signature de l'acheteur', 'Signature du vendeur', 'Deuxième témoin', and 'Premier témoin'. At the bottom of the interface, there are two buttons: 'retour' (return) and 'imprimez' (print).

Figure 20: Interface de Notaire

Lors de l'entrée des informations, nous cliquons pour confirmer ces informations. Elles apparaissent ensuite sous la forme d'un document contenant toutes les informations que nous avons précédemment saisies. Dans ce document, apparaissent les signatures : de l'acheteur, du vendeur, du premier témoin et du deuxième témoin

3.Conclusion :

En somme, ce chapitre nous a permis de rendre visible les aspects qui ont été évoqués dans le chapitre précédent. Il correspond à la dernière partie de ce rapport et a pour objet de clarifier toutes les fonctionnalités de l'application.

Conclusion Générale :

Au cours de ce travail, j'ai présenté les différentes étapes qui ont conduit à la mise en œuvre de ce système de gestion pour le bureau des notaires. J'ai commencé par identifier les difficultés rencontrées par les notaires et les clients dans les bureaux de notaires afin de proposer une solution adéquate et de définir les besoins. Le langage de modélisation UML et le processus unifié UP ont servi de soutien à l'analyse des besoins et à la conception de l'application web à travers les différents diagrammes UML couvrant les aspects fonctionnels, dynamiques et statiques du développement global.

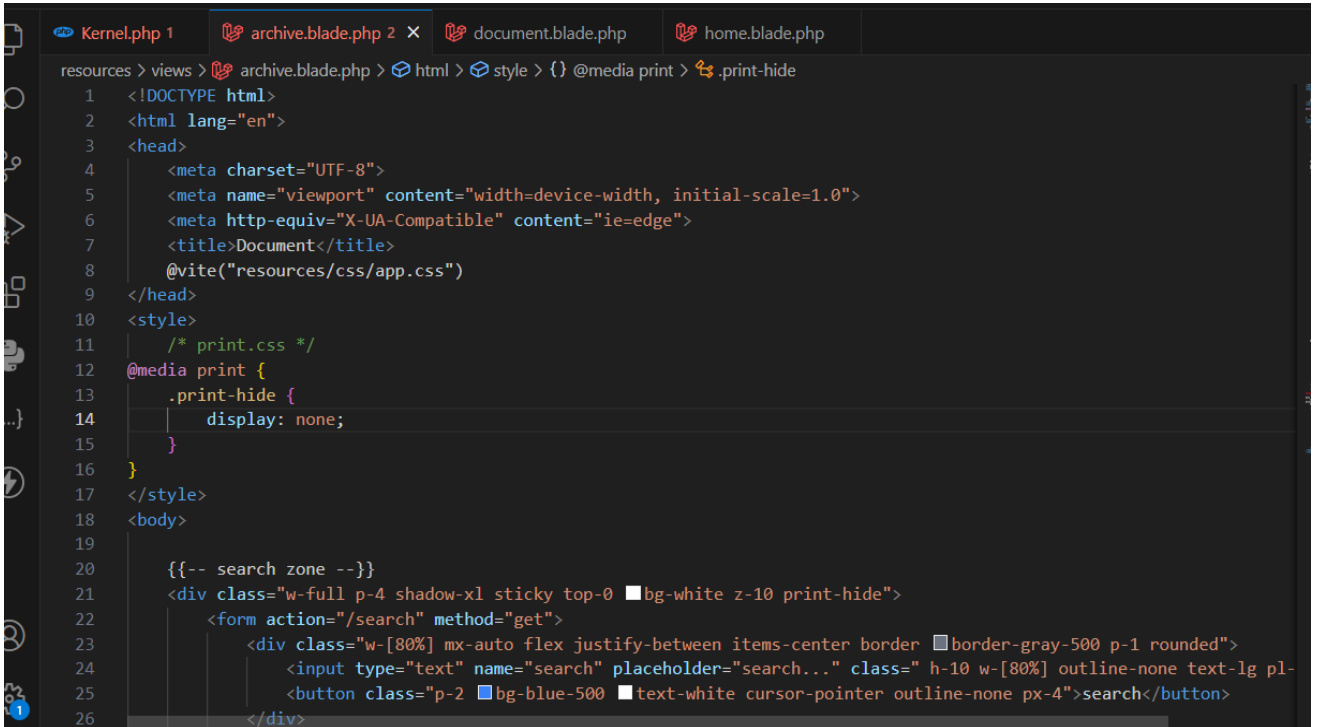
Pour la mise en œuvre de l'application, j'ai utilisé le SGBD MYSQL pour la base de données et les frameworks Laravel (PHP) et Bootstrap ainsi que les langages HTML et CSS. En conclusion, ce projet a été une expérience enrichissante et très utile pour moi. En effet, il m'a permis d'enrichir mes connaissances théoriques et mes compétences en matière de conception et de programmation. De plus, j'ai pu appliquer les connaissances acquises tout au long de mes études.

Par ailleurs, j'ai constaté que le projet de développement de logiciels n'est jamais totalement terminé, c'est pourquoi nous prévoyons d'apporter des améliorations supplémentaires à l'application à l'avenir.

○ Bibliographie

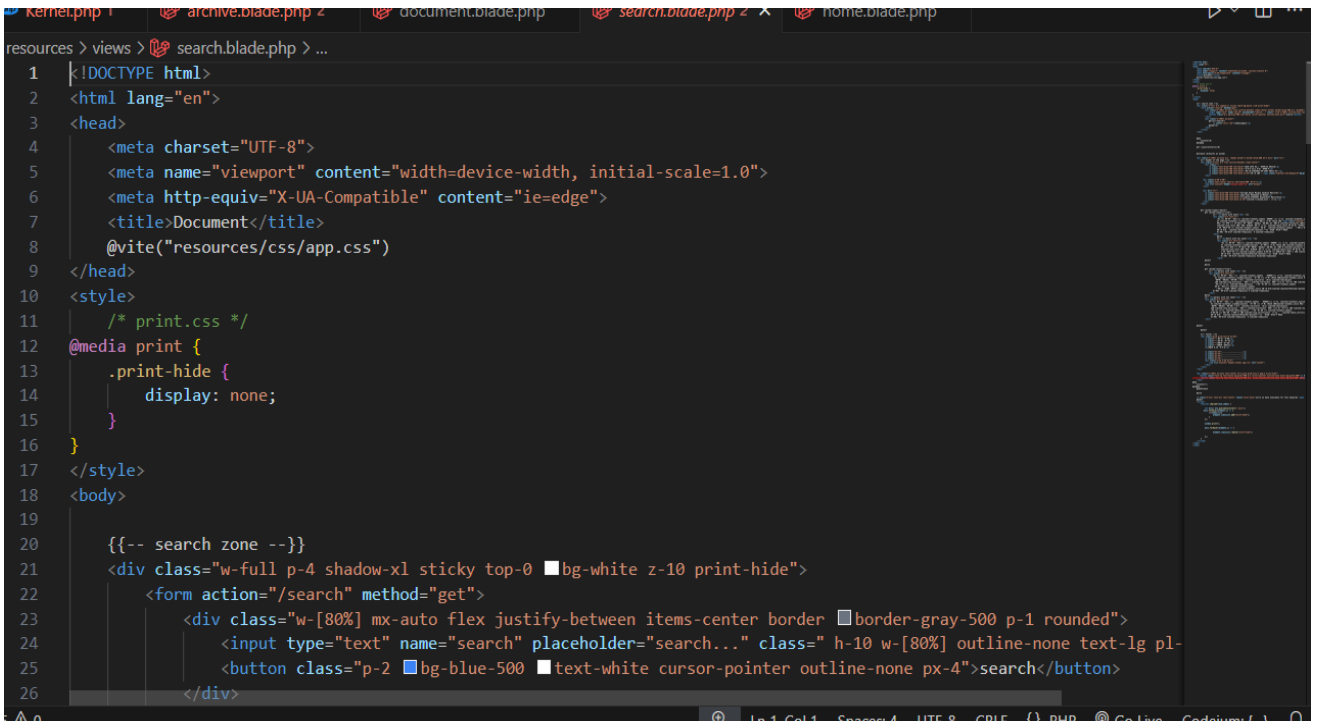
- Boot Docs.* (2024, 08 15). Récupéré sur Bootstrap: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- CSS Documentation.* (2024, 08 17). Récupéré sur CSS : <https://developer.mozilla.org/en-US/docs/Web/CSS>
- (2024). *Gestion Bibliotheque*. Nouakchott: ISCAE.
- HTML Introduction.* (2024, 08 27). Récupéré sur w3schools: https://www.w3schools.com/html/html_intro.asp
- Introduction.* (2024, 08 19). Récupéré sur mozilla: https://developer.mozilla.org/fr/docs/Learn_web_development/Howto/Web_mechanics/What_is_a_web_server
- Laravel.* (2024, 08 15). Récupéré sur laravel: <https://laravel.com/>
- PHP Docs.* (2024, 08 15). Récupéré sur PHP: <https://www.youtube.com/watch?v=KBT2gmAfav4>
- SQL Docs.* (2024, 08 20). Récupéré sur w3schools: <https://www.w3schools.com/sql/>
- VS Docs.* (2024, 08 15). Récupéré sur VScode: <https://code.visualstudio.com/docs>

Annexes



```
resources > views > archive.blade.php > html > style > {} @media print > .print-hide
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title>Document</title>
8     @vite("resources/css/app.css")
9 </head>
10 <style>
11     /* print.css */
12 @media print {
13     .print-hide {
14         display: none;
15     }
16 }
17 </style>
18 <body>
19
20     {{-- search zone --}}
21     <div class="w-full p-4 shadow-xl sticky top-0 bg-white z-10 print-hide">
22         <form action="/search" method="get">
23             <div class="w-[80%] mx-auto flex justify-between items-center border border-gray-500 p-1 rounded">
24                 <input type="text" name="search" placeholder="search..." class="h-10 w-[80%] outline-none text-lg pl-
25                 <button class="p-2 bg-blue-500 text-white cursor-pointer outline-none px-4">search</button>
26             </div>
```

Figure 21: Archive.php



```
resources > views > search.blade.php > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title>Document</title>
8     @vite("resources/css/app.css")
9 </head>
10 <style>
11     /* print.css */
12 @media print {
13     .print-hide {
14         display: none;
15     }
16 }
17 </style>
18 <body>
19
20     {{-- search zone --}}
21     <div class="w-full p-4 shadow-xl sticky top-0 bg-white z-10 print-hide">
22         <form action="/search" method="get">
23             <div class="w-[80%] mx-auto flex justify-between items-center border border-gray-500 p-1 rounded">
24                 <input type="text" name="search" placeholder="search..." class="h-10 w-[80%] outline-none text-lg pl-
25                 <button class="p-2 bg-blue-500 text-white cursor-pointer outline-none px-4">search</button>
26             </div>
```

Figure 22: Search.php

```

Kernel.php 1  archive.blade.php 2  document.blade.php  search.blade.php 2 x  home.blade.php
resources > views > search.blade.php > ...
2  <html lang="en">
18 <body>
20  {{-- search zone --}}
21  <div class="w-full p-4 shadow-xl sticky top-0 bg-white z-10 print-hide">
22    <form action="/search" method="get">
23      <div class="w-[80%] mx-auto flex justify-between items-center border border-gray-500 p-1 rounded">
24        <input type="text" name="search" placeholder="search..." class="h-10 w-[80%] outline-none text-lg pl-
25        <button class="p-2 bg-blue-500 text-white cursor-pointer outline-none px-4">search</button>
26      </div>
27      <div class="w-[80%] mx-auto">
28        @error("search")
29          <p style="color:red">{{ $message }}</p>
30        @enderror
31      </div>
32    </form>
33  </div>
34
35
36  @php
37    $counter=0;
38  @endphp
39
40  @if (count($results)>0)
41
42

```

Figure 23: Search.php

```

2014_10_12_000000_create_users_table.php  database.php  Document.php x  User.php  2024_03_06_191935_create_documents_ta
app > Models > Document.php > Document
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Document extends Model
9  {
10     protected $fillable = [
11         "vendeur_name",
12         "acheteur_name",
13         "vendeur_nni",
14         "acheteur_nni",
15         "acheteur_naissance",
16         "acheteur_lieuNaissance",
17         "acheteur_gender",
18         "secteur",
19         "num_sect",
20         "num_res",
21         "cartier",
22         "belidiya",
23         "num_terrain",
24         "surface",
25         "num_permis",
26         "date_permis",
27         "num_certificat",
28         "date_certificat",
29         "nom_imam",

```

Figure 24: Document.php