

## **DÉDICACE**

À

Mon père KOLANI Natiéyendou, Ma Mère YOMBOU Sanibile, Mon Frère KOLANI  
Pouguinimpo et toutes les personnes impliquées dans mon parcours.

## REMERCIEMENT

Je tiens à exprimer ma profonde gratitude envers plusieurs personnes et entités qui ont joué un rôle essentiel dans la réussite de mon stage et dans la réalisation de ce mémoire. Leur soutien inestimable et leurs encouragements m'ont été d'une aide précieuse tout au long de cette expérience enrichissante.

Mes sincères remerciements vont à l'endroit de :

- **Dr TIEBEKABE**, Responsable pédagogique de la licence professionnelle en Sécurité informatique et Cybersécurité, pour son dévouement envers la réussite de ses étudiants ;
- **Dr BASSAGOU**, Docteur Informaticien, Enseignant chercheurs à l'université de KARA, pour son accompagnement précieux, ses conseils avisés, et sa disponibilité tout au long de ce travail. Je lui suis particulièrement reconnaissant pour la confiance qu'il m'a accordée, ainsi que pour les encouragements constants qui ont été déterminants pour mener à bien ce projet. Sa passion pour le sujet a été une véritable source de motivation, et je le remercie sincèrement pour son soutien et ses précieuses orientations ;
- **M. KPATCHA**, Enseignant à l'université de KARA, pour Sa disponibilité et ces conseils précieux ;
- **M. Komla Dodji KOMLATSE**, le directeur général de Nova Lead, pour son encadrement attentif et ses précieux conseils tout au long de mon stage. Son expertise, sa disponibilité, et sa volonté de partager ses connaissances ont été d'une grande aide dans mon apprentissage. Je le remercie pour la confiance qu'il m'a accordée et pour m'avoir offert l'opportunité de développer mes compétences et surtout acquérir une expérience dans un environnement professionnel enrichissant ;
- **Tout le corps enseignant de l'UK**, pour les enseignements précieux et la guidance exceptionnelle qu'ils ont offerts tout au long de notre parcours académique ;
- **AYA Yawavi Etsiam, Romaric, Emmanuel ADOKO, Athanase** Pour leurs soutiens moraux et leur motivation
- **Toute la famille**, pour vos soutiens. Ce mémoire est le fruit de vos encouragements et de votre confiance en moi.

Enfin, je tiens à remercier le cabinet NOVA LEAD pour avoir mis à ma disposition les ressources nécessaires pour mener à bien ce stage.

## RÉSUMÉ

Nova Lead, une entreprise spécialisée dans les logiciels sur mesure, joue un rôle central dans le développement et l'innovation technologique. Pour répondre aux défis spécifiques liés à la gestion de l'assistance client, notamment le suivi des demandes, la résolution des problèmes et l'optimisation de la communication, une solution personnalisée a été développée.

Cette plateforme repose sur une approche orientée objet, modélisée à l'aide du langage UML pour structurer et gérer efficacement les différents aspects du système. Le projet a été conçu avec Vue.js pour l'interface utilisateur, JavaScript pour la programmation côté client, et Node.js pour le développement du backend. MongoDB, une base de données orientée documents, a été sélectionnée pour sa flexibilité et sa capacité à gérer les données complexes de manière efficace. La plateforme permet la création et la gestion centralisée des tickets. Chaque demande client est transformée en ticket, ce qui facilite le suivi et la résolution des problèmes. Les détails de chaque demande, comme le nom du client, le type de demande, le produit concerné, le statut et la date de création, sont stockés et accessibles en un seul endroit. En outre, la communication entre les équipes et les clients est optimisée grâce à un système de messagerie intégré, rendant les échanges plus fluides et mieux organisés.

Cette solution améliore considérablement la visibilité des interactions avec les clients, renforce la collaboration entre les équipes et optimise globalement la gestion des demandes d'assistance. Nova Lead envisage également d'étendre cette plateforme en y intégrant un module de gestion de la prospection. Cet ajout permettra de suivre les prospects, de centraliser les interactions commerciales et de renforcer la gestion de la relation client, ouvrant ainsi de nouvelles opportunités de croissance et de développement pour l'entreprise.

## ABSTRACT

Nova Lead, a company specializing in custom software, plays a central role in technological development and innovation. To address specific challenges related to customer support management, including tracking requests, resolving issues, and optimizing communication, a tailored solution has been developed.

This platform is based on an object-oriented approach, modeled using UML to effectively structure and manage various aspects of the system. The project was designed with Vue.js for the user interface, JavaScript for client-side programming, and Node.js for backend development. MongoDB, a document-oriented database, was chosen for its flexibility and ability to handle complex data efficiently.

The platform enables the centralized creation and management of tickets. Each customer request is transformed into a ticket, making it easier to track and resolve issues. Details of each request, such as the client's name, type of request, related product, status, and creation date, are stored and accessible in one place. Additionally, communication between teams and clients is optimized through an integrated messaging system, ensuring smoother and better-organized exchanges.

This solution significantly enhances visibility into customer interactions, strengthens team collaboration, and improves overall management of support requests. Nova Lead also plans to expand this platform by integrating a prospect management module. This addition will allow for tracking prospects, centralizing commercial interactions, and strengthening customer relationship management, opening new opportunities for growth and development for the company.

## SOMMAIRE

DÉDICACE .....	1
REMERCIEMENT .....	2
RÉSUMÉ .....	3
ABSTRACT.....	4
SOMMAIRE .....	5
LISTE DES TABLEAUX .....	7
LISTE DES FIGURES.....	8
LISTE DES PARTICIPANTS .....	9
GLOSSAIRE.....	10
INTRODUCTION GÉNÉRALE .....	11
CHAPITRE 1 : PRÉSENTATION DU CADRE DE STAGE.....	12
1.1 Présentation de l'entreprise d'accueil : Nova Lead.....	12
CHAPITRE 2 : CAHIER DES CHARGES.....	14
2.1 Contexte et justification du projet .....	14
2.2 Problématique .....	14
2.3 Objectifs .....	15
2.4 Résultats attendus.....	15
2.5 Étude et critique de l'existant.....	16
2.6 Spécifications fonctionnelles .....	16
2.7 Spécifications non fonctionnelles .....	17
2.8 Proposition et choix de solution.....	18
2.9 Planning de réalisation du projet.....	21
CHAPITRE 3 : ANALYSE ET CONCEPTION.....	22

4.1 Présentation de la méthode d'analyse et de conception .....	22
4.2 Présentation de l'outils de modélisation .....	26
4.3 Etude détaillé de la solution.....	27
CHAPITRE 4 : RÉALISATION ET MISE EN ŒUVRE .....	42
7.1 Mise en œuvre .....	42
7.2 Présentation de l'application .....	49
CONCLUSION GÉNÉRALE.....	58
Bibliographie.....	59
TABLE DES MATIÈRES .....	I

## **LISTE DES TABLEAUX**

Tableau 1 : Participants du projet .....	9
Tableau 2 : Glossaire .....	10
Tableau 3 : Planning prévisionnel de réalisation du projet.....	21
Tableau 4 : Description textuelle du cas d'utilisation " enregistrer un client " .....	33
Tableau 5 : Description textuelle du cas d'utilisation « enregistrer un produit ».....	34
Tableau 6 Description textuelle du cas d'utilisation « enregistrer un nouveau type de demande » .....	35
Tableau 7 : Description textuelle du cas d'utilisation « enregistrer un Ticket » .....	36
Tableau 8 : Description textuelle du cas d'utilisation « enregistrer une note ».....	37
Tableau 9 : Caractéristiques du matériel de développement .....	42
Tableau 10 : Coût financier du projet .....	49

## LISTE DES FIGURES

Figure 1 : Localisation géographique avec GPS de Nova Lead.....	13
Figure 2 : Logo de UML .....	23
Figure 3 : Schématisation du modèle en V .....	24
Figure 4 : Logo de Draw.io.....	26
Figure 5 : : Diagramme du contexte statique .....	28
Figure 6 : Diagramme des cas d'utilisation « Administrateur » .....	30
Figure 7 : Diagramme des cas d'utilisation « Agent Support ».....	31
Figure 8 : Diagramme des cas d'utilisation « Client » .....	32
Figure 9 : Diagramme d'activité du cas d'utilisation « enregistrer un client » .....	38
Figure 10 : Diagramme d'activité du cas d'utilisation « enregistrer un ticket » .....	39
Figure 11 : Diagramme d'activité du cas d'utilisation « ajouter une note » .....	40
Figure 12 : Diagramme des classes du système de gestion de Nova Lead .....	41
Figure 13 : modèle MVC.....	46
Figure 14 : Structure de l'application – espace Administrateur .....	50
Figure 15 : Structure de l'application – espace d'Agent Support .....	51
Figure 16 : Structure de l'application – espace Client .....	52
Figure 17 : Architecture de l'application.....	53
Figure 18 : Interface de saisie de données de client par l'administrateur .....	54
Figure 19 : Interface de saisie de données des produits .....	55
Figure 20 : Interface de saisie de données des types de demande.....	55
Figure 21 : Interface de saisie de données d'ajout de tickets .....	56
Figure 22 : interface d'échange entre le client et l'administrateur ou l'agent support.....	57



## LISTE DES PARTICIPANTS

*Tableau 1 : Participants du projet*

Nom et prénoms	Poste	Rôle
KOLANI Florent	Etudiant en LPDM	Concepteur et réalisateur du projet
Dr Dikagma BASSAGOU	Enseignant à L'UK	Directeur de mémoire
Mr KOMLATSE Komla Dodji	Directeur de NOVA LEAD	Maitre de stage

## GLOSSAIRE

Tableau 2 : Glossaire

Terme	Définition
LPDM	Licence Professionnelle en Développement web et Mobile, le centre de formation dans lequel j'ai été formée.
Nova Lead	L'entreprise dans laquelle j'ai fait mon stage.
CRM	Customer Relationship Management
UML	UML (Unified Modeling Language) est un langage de modélisation visuelle standardisé utilisé pour spécifier, concevoir, documenter et visualiser les aspects d'un système logiciel.
API	Une API (Application Programming Interface), est un programme permettant à deux applications distinctes de communiquer entre elles.
UK	Université de KARA

## INTRODUCTION GÉNÉRALE

Dans un contexte économique où la digitalisation s'impose comme une nécessité, les entreprises doivent optimiser leurs processus pour rester compétitives. Nova Lead, en pleine expansion, a identifié un besoin essentiel d'améliorer la gestion de son support client et de ses flux de communication. En effet, cette croissance rapide a entraîné une complexification des opérations, rendant difficile le suivi des demandes d'assistance, la gestion efficace des réponses aux clients et le maintien d'un suivi rigoureux des interactions et des résolutions.

Ce projet propose la création d'une plateforme d'assistance en ligne pour répondre à ces défis. L'objectif de cette plateforme est de standardiser et d'optimiser le traitement des demandes de support, depuis l'ouverture d'un ticket jusqu'à sa résolution, afin de renforcer l'efficacité et la précision du service client de Nova Lead. Le thème retenu est donc : la mise en place d'un système de gestion des demandes de support en ligne et des flux de communication d'une entreprise, cas de Nova Lead.

Ce mémoire, structuré en quatre (4) chapitres, décrit les étapes de réalisation de ce projet mené dans le cadre de mon stage de fin de formation. Le premier chapitre présente le contexte du lieu de stage, tandis que le deuxième chapitre est consacré au cahier des charges du projet, en détaillant la problématique rencontrée et l'intérêt de développer une solution adaptée. Le troisième chapitre se concentre sur l'analyse et la conception de la solution, en exposant la méthode d'analyse utilisée, les outils de modélisation choisis, ainsi qu'une étude approfondie de la solution envisagée. Enfin, le quatrième chapitre traite des moyens et outils mis en œuvre, ainsi que des résultats obtenus.

## CHAPITRE 1 : PRÉSENTATION DU CADRE DE STAGE

Ce chapitre présente l'institution qui a joué un rôle déterminant dans mon parcours professionnel : l'entreprise Nova Lead. Cette organisation a grandement contribué à l'enrichissement de mes compétences et à l'élargissement de ma vision professionnelle. Les statuts, les missions, les activités de Nova Lead, ainsi que certaines de ses réalisations notables, seront explorés en détail.

### 1.1 Présentation de l'entreprise d'accueil : Nova Lead

Nova Lead est une entreprise de production de logiciels sur mesure, spécialisée dans le domaine de la microfinance. Depuis sa création, elle s'engage à fournir des solutions financières innovantes et adaptées aux besoins des petites et moyennes entreprises ainsi qu'aux particuliers.

(1)

#### 1.2.1 Mission

Concevoir des solutions logicielles sur mesure qui facilitent la gestion et le développement des entreprises, en leur offrant des outils numériques innovants, fiables et adaptés à leurs besoins spécifiques forment la mission de Nova Lead.

#### 1.2.2 Activités

Les activités faites par Nova Lead sont suivantes :

- **Le développement de logiciels sur mesure** : la conception et le développement de logiciels personnalisés pour répondre aux besoins spécifiques des clients dans divers secteurs, mais surtout, la microfinance.
- **Les services de maintenance et de support technique** : la fourniture de services de maintenance continue et de support technique pour assurer la performance, la fiabilité et la sécurité des solutions logicielles livrées.
- **L'hébergement des applications** : l'hébergement des solutions logicielles dans des environnements sécurisés, en assurant une gestion efficace des serveurs, des bases de données et des services cloud.

- **La vente de matériel informatique indispensable à l'utilisation de ses productions :**

La vente de matériel informatique complémentaires pour assurer une utilisation optimale des logiciels développés.

### 1.2.3 Quelques réalisations

Nova Lead a déjà conçu des logiciels parmi lesquels peuvent être cités :

- **DOGA KABA** : une plateforme digitale de gestion des demandes de crédit, des plaintes, des réclamations, de la performance sociale et environnementale. Elle est spécialement conçue sur mesure pour les microfinances ; (2)
- **LETAVI** : une plateforme de services à valeur ajoutée (envoi de SMS en masse) ; (3)
- **MIVO CRM** : une plateforme de décentralisation des activités opérationnelles de MIVO ENERGY. (4)

### 1.2.4 Plan de localisation

Nova Lead se situe à Agbalepedogan, non loin de la pharmacie Matilda, spécialement à Open the box. Le plan de localisation de Nova Lead fourni par google map (5).

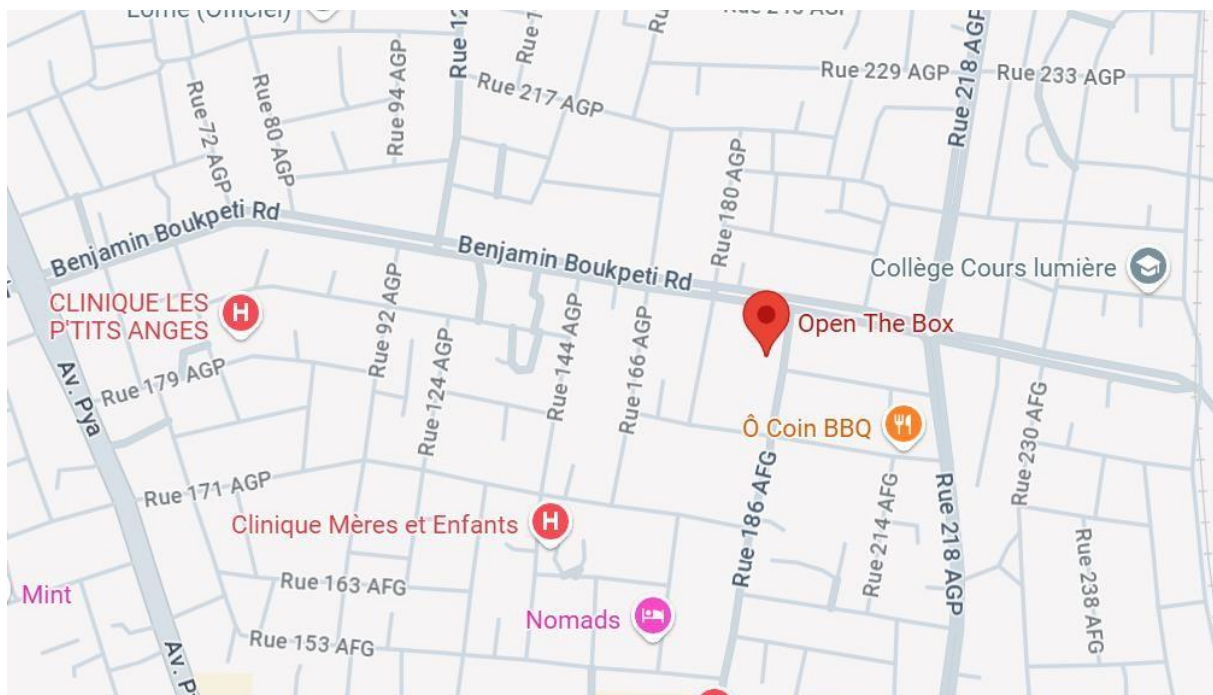


Figure 1 : Localisation géographique avec GPS de Nova Lead

## **CHAPITRE 2 : CAHIER DES CHARGES**

Le cahier des charges est un document essentiel qui formalise les besoins, objectifs et les contraintes du projet. Il sert de référence tout au long du développement, garantissant que les attentes des parties prenantes soient correctement traduites en spécifications fonctionnelles et techniques. Ce chapitre expose le cahier des charges du projet ainsi que son importance dans la réussite de celui-ci.

### **2.1 Contexte et justification du projet**

Nova Lead est une entreprise spécialisée dans la production de logiciels sur mesure et de progiciels, offrant un accompagnement complet tout au long du cycle de vie de ses produits, y compris la maintenance et les mises à jour. L'entreprise conçoit des solutions logicielles personnalisées ainsi que des services complémentaires, garantissant une utilisation optimale de ses produits.

Dans un contexte de croissance continue et face à un volume croissant de demandes de support client, Nova Lead rencontre des défis significatifs liés à la gestion des demandes d'assistance et au suivi des résolutions. Les méthodes actuelles, souvent basées sur des outils fragmentés et des processus manuels, engendrent des retards et des incohérences dans le suivi, ce qui limite l'efficacité de l'entreprise et sa capacité à répondre rapidement aux besoins de ses clients. Pour relever ces défis, Nova Lead souhaite mettre en place une plateforme d'assistance en ligne permettant d'optimiser le traitement des demandes, d'assurer un suivi structuré des interactions et d'améliorer la qualité du service client.

### **2.2 Problématique**

NOVA LEAD utilise actuellement des méthodes manuelles et dispersées pour gérer les demandes d'assistance, ce qui entraîne des inefficacités. Les tickets d'assistance sont gérés dans divers fichiers et feuilles de calcul, ce qui complique le suivi des demandes et des résolutions. Les réponses et les interactions sont parfois fragmentées, et l'absence d'un système centralisé limite la visibilité globale sur les opérations d'assistance. Cela entraîne des retards dans le suivi des demandes et affecte la satisfaction client.

Suite aux constats, naissent trois questions clés suivantes :

- Comment centraliser et automatiser la gestion des tickets pour un suivi efficace des demandes des clients ?
- Quels processus de catégorisation et de priorisation des tickets pourraient optimiser les réponses et les délais de résolution ?
- Comment une plateforme digitale d'assistance pourrait-elle améliorer la qualité du support et la satisfaction client ?

## **2.3 Objectifs**

### **2.3.1 Objectif général**

L'objectif général est de créer une plateforme d'assistance en ligne qui centralise les demandes des clients, les suivis et les résolutions

### **2.3.2 Objectif spécifique**

Spécifiquement, le projet vise à :

- D'améliorer la gestion des demandes en centralisant et en standardisant le suivi des tickets d'assistance.
- D'accélérer la résolution des tickets en automatisant les tâches de catégorisation et d'assignation.
- D'offrir une vue globale et des rapports sur les performances de l'assistance.

## **2.4 Résultats attendus**

Au terme de ce projet, les principaux résultats attendus se résument au développement d'un système couvrant les principaux processus d'assistance de l'entreprise, notamment :

- La gestion des clients et de leurs demandes d'assistance,
- Le suivi et la résolution des tickets de support,
- L'amélioration de la communication et de la traçabilité des interactions avec les clients.

## **2.5 Étude et critique de l'existant**

L'analyse des systèmes existants est essentielle pour identifier les insuffisances et opportunités d'amélioration. Cette étape fournit une base solide pour une critique constructive et une conception innovante.

### **2.5.1 Étude de l'existant**

L'entreprise gère les informations de ses clients principalement via WhatsApp et par e-mail. Le suivi des interactions avec les clients repose sur des échanges dispersés dans les discussions et les fils de messagerie, rendant difficile l'accès rapide à l'historique des demandes.

### **2.5.2 Critique de l'existant**

Les méthodes actuelles présentent plusieurs limites. La gestion des données clients via WhatsApp et e-mail est vulnérable aux pertes d'information et rend difficile le suivi centralisé des demandes. La dispersion des échanges rend la coordination complexe et complique la vision globale des interactions avec chaque client. En l'absence de système centralisé, le suivi des demandes et des résolutions est chronophage (demande plus de temps) et peut entraîner des incohérences, affectant la réactivité de l'entreprise et la qualité du service client.

## **2.6 Spécifications fonctionnelles**

Le système de gestion des demandes d'assistance en ligne de Nova Lead introduit une automatisation et une simplification des processus essentiels, afin de rendre les opérations de l'entreprise plus fluides et précises. Il propose des fonctionnalités couvrant plusieurs aspects, de la gestion des clients et des demandes de support à la gestion des utilisateurs et du suivi des résolutions des tickets. Voici un aperçu détaillé de ce que le système permettra de réaliser.

### **2.6.1 Gestion des clients**

Le système permettra d'enregistrer, de modifier, d'archiver, de désarchiver un client et de consulter la liste de tous les clients, classée par état (archivé ou désarchivé). La liste sera filtrable par nom, par type (moral ou physique) et par pays. De plus, il offrira une vue détaillée de l'historique des demandes d'assistance et des tickets associés à chaque client.



## 2.6.2 Gestion de demandes d'assistance

Le système permettra d'enregistrer, de modifier, d'archiver et de désarchiver une demande d'assistance (ticket) et de consulter la liste de tous les tickets ouverts. La liste sera filtrable par statut (en cours, résolu, ou clos), par type de demande (technique, commercial, etc.), et par client. Les tickets pourront également être triés par priorité et date d'ouverture.

## 2.6.3 Suivi des tickets et communication

Le système permettra de suivre l'état d'un ticket en temps réel, d'ajouter des commentaires ou des messages entre le client et l'équipe de support, et de clôturer un ticket une fois la demande résolue. Il offrira également un tableau de bord permettant aux utilisateurs (administrateurs et agents de support) de visualiser tous les tickets en attente, résolus ou clôturés, avec un accès facile aux détails de chaque demande.

## 2.6.4 Gestion des utilisateurs

Le système permettra de créer, modifier, archiver et désarchiver des utilisateurs (agents de support, administrateurs et clients). Il permettra également de définir des rôles pour chaque utilisateur (par exemple, support, admin) et de gérer les permissions associées à chaque rôle pour garantir un accès approprié aux fonctionnalités du système.

## 2.6.5 Suivi et rapports

Le système permettra de générer des rapports détaillés sur les tickets ouverts, en cours, résolus et fermés, ainsi que sur les interactions entre les clients et les agents. Ces rapports aideront à analyser les performances de l'équipe de support et à identifier les domaines à améliorer.

## 2.7 Spécifications non fonctionnelles

Les spécifications non fonctionnelles du système incluent :

- **Compatibilité multi-plateforme** : Le système doit être accessible depuis différentes plateformes, telles que les ordinateurs de bureau, les tablettes et les smartphones, et s'adapter de manière responsive à toutes tailles d'écran.

- **Ergonomie** : Le système doit offrir une expérience utilisateur fluide et intuitive, afin de réduire le temps d'apprentissage et d'améliorer la productivité des utilisateurs. L'interface doit être claire, simple, et facile à naviguer.
- **Sécurité et confidentialité** : Le système doit intégrer des mesures de sécurité rigoureuses pour protéger les données sensibles des clients et de l'entreprise. Cela inclut des mécanismes d'authentification sécurisés, le chiffrement des données sensibles, et des contrôles d'accès basés sur les rôles pour assurer la confidentialité et la sécurité des informations.

## 2.8 Proposition et choix de solution

Avec les objectifs et les spécifications du projet désormais bien définis, il est temps d'évaluer les solutions disponibles. Dans cette section, deux principales solutions sont proposées à Nova Lead, qui sont décrites en détail ci-dessous.

### 2.8.1 Première solution : Adoption d'un CRM

Cette première proposition consiste à adopter une solution CRM (Customer Relationship Management) existante, telle que Sellsy ou Odoo. Ces plateformes offrent des fonctionnalités étendues pour la gestion des interactions client, le suivi des demandes d'assistance, ainsi que des outils supplémentaires pour la facturation et la comptabilité. Une telle solution permettrait de couvrir une grande partie des besoins de Nova Lead en matière de support client et de gestion des flux de travail.

#### 2.8.1.1 Sellsy

Sellsy est une solution logicielle qui centralise la gestion commerciale et les opérations financières d'une entreprise (6). Elle propose des fonctionnalités de gestion de la relation client (CRM), de suivi des interactions client, de facturation, de gestion des devis, et de relance des paiements. Son interface intuitive, ainsi que ses intégrations avec des applications populaires telles qu'Office 365 et PayPal, permettent de simplifier et d'optimiser le suivi des demandes et la facturation client, offrant ainsi une solution clé en main pour la gestion de la relation client.

### 2.8.1.2 Odoo

Odoo est un ERP open-source très flexible, offrant une suite complète de modules pour divers processus d'entreprise, dont la gestion de la relation client, le suivi des interactions, et la facturation. Odoo propose une version communautaire gratuite et une version payante pour les entreprises nécessitant des fonctionnalités avancées. Sa modularité et sa forte capacité de personnalisation permettent de configurer et d'adapter les fonctionnalités aux besoins de Nova Lead, tout en assurant une gestion centralisée et fluide des informations client, avec un support de communauté pour la version gratuite et un support officiel pour la version entreprise.

### 2.8.2 Deuxième solution : Développement d'une application sur mesure

Cette solution propose de concevoir et de développer une application web de gestion d'assistance sur mesure, spécifiquement adaptée aux besoins de Nova Lead. Cette application permettrait de centraliser et d'automatiser les processus essentiels, notamment la gestion des tickets de support, le suivi des interactions client. En créant une solution sur mesure, Nova Lead bénéficierait d'un outil qui répond parfaitement aux spécifications et aux flux de travail propres à l'entreprise, avec une grande flexibilité pour l'évolution future.

### 2.8.3 Choix de Solution

Bien que l'adoption d'une plateforme CRM existante comme Sellsy ou Odoo présente plusieurs avantages :

- **Rapidité de mise en place** : Le déploiement de ces solutions est rapide, permettant une adoption en un temps réduite.
- **Modules variés** : Ces solutions proposent une gamme de modules pour la gestion des clients, des devis, des factures et du suivi des interactions, répondant ainsi à divers besoins fonctionnels.

Elles présentent cependant certains inconvénients notables :

- **Facturation pour modules inutilisés** : Bien que ces plateformes incluent de nombreux modules, tous ne sont pas forcément pertinents pour Nova Lead. Cependant, le coût d'abonnement inclut souvent l'ensemble des fonctionnalités, y compris celles qui ne seront pas utilisées, ce qui augmente le coût global.

- **Limite de contrôle et d'évolutivité** : La dépendance à un fournisseur externe réduit la capacité de Nova Lead à personnaliser et à faire évoluer le système en fonction de ses besoins spécifiques et de sa croissance.

En revanche, la deuxième solution, qui consiste en une application sur mesure, permettrait de créer un système parfaitement adapté aux besoins spécifiques de Nova Lead. Cette approche offre :

- **Personnalisation** : Un contrôle total sur les fonctionnalités, les interfaces et les flux de travail.
- **Adaptabilité** : La possibilité de faire évoluer l'application en fonction des besoins futurs et des retours des utilisateurs, sans dépendre de contraintes externes.

Toutefois, cette solution sur mesure comporte également certains inconvénients :

- **Temps de mise en œuvre** : La conception et le développement nécessitent plusieurs mois, retardant ainsi la mise en production.
- **Coût initial** : Le développement d'une solution sur mesure requiert un investissement initial plus élevé, en raison des ressources humaines et techniques nécessaires.

Après analyse, la solution sur mesure semble être la plus avantageuse pour Nova Lead. Elle permettra d'obtenir un outil de gestion conçu spécifiquement pour les besoins de l'entreprise, avec une adaptabilité optimale pour les équipes et un contrôle total sur les fonctionnalités.

## 2.9 Planning de réalisation du projet

Le tableau suivant détaille le planning de réalisation du projet.

Tableau 3 : Planning prévisionnel de réalisation du projet

Phase	Tâche		Date de début	Date de fin	Durée (jr)
1	Analyse des besoins et conception d'une solution		02/09/24	09/09/24	7
2	Modélisation (UML)		10/09/24	17/09/24	7
3	Conception de design UI		18/09/24	25/09/24	7
4	Développement	Implémentation d'API	26/09/24	15/10/24	20
		Implémentation d'interface graphique	16/10/24	01/11/24	15
		Intégration et test unitaire	02/11/24	12/11/24	10
5	Test et correction de bug		13/11/24	28/11/24	15
6	Déploiement		29/11/24	02/12/24	4
Date de fin de projet			02/12/24	Total	85

## **CHAPITRE 3 : ANALYSE ET CONCEPTION**

À ce stade du projet, après la définition des objectifs et des exigences, il est essentiel de passer à l'analyse et à la conception. Cette phase permet de traduire les besoins identifiés en solutions techniques concrètes. L'analyse vise à décomposer les problèmes, à évaluer les contraintes, et à identifier les meilleures approches pour les résoudre. La conception, quant à elle, consiste à élaborer une architecture claire et structurée qui servira de fondation au développement. Ce chapitre détaille alors, les méthodes et outils utilisés pour analyser les données et concevoir une solution adaptée aux enjeux du projet.

### **4.1 Présentation de la méthode d'analyse et de conception**

Il est primordial de choisir une méthode d'analyse et de conception adaptée aux exigences du projet. Cette étape permet d'établir une base solide pour toutes les phases ultérieures du développement. L'analyse rigoureuse des besoins et la conception structurée sont des éléments clés pour garantir la réussite du projet. Dans cette section, l'accent sera mis sur les méthodes d'analyse et de conception qui ont guidé les différentes étapes de ce projet.

#### **4.1.1 Méthode d'analyse**

L'analyse des systèmes peut être abordée de différentes manières. L'approche procédurale se concentre sur les processus et les flux de données, décomposant le système en une série de procédures séquentielles, ce qui est particulièrement utile pour comprendre les processus métiers et les interactions au sein du système. En revanche, l'approche orientée objet modélise le système à travers des objets qui intègrent à la fois les données et les comportements, utilisant des concepts tels que les classes, les objets, et leurs relations pour offrir une représentation plus modulaire et réutilisable. Pour ce projet, l'approche orientée objet a été retenue, car elle permet de créer un modèle extensible et modulaire, idéal pour gérer des exigences complexes et s'adapter aux évolutions futures. Cette approche, supportée par l'outil UML, permet de modéliser efficacement les solutions en offrant une vue structurée et intégrée du système envisagé.

#### 4.1.1.1 Présentation du langage de modélisation UML

UML (**Langage de modélisation unifié**) est un langage de modélisation standardisé utilisé pour visualiser, spécifier, construire, et documenter les éléments d'un système logiciel (7). Il offre un ensemble de diagrammes permettant de représenter les aspects statiques, dynamiques, et comportementaux d'un système, facilitant ainsi la compréhension et la communication entre les différentes parties prenantes du projet. UML est largement utilisé pour modéliser des systèmes orientés objet, en capturant les concepts clés tels que les classes, les objets, les interactions, et les états, offrant ainsi une vue claire et organisée de l'architecture du système. Ce langage est celui utilisé pour la modélisation et l'étude détaillée du projet.



*Figure 2 : Logo de UML*

#### 4.1.2 Méthode de conception

La méthode de conception est cruciale pour transformer les exigences définies en une solution fonctionnelle et structurée. Dans ce projet, une approche combinée du modèle en V a été adoptée. Il est une méthode de conception rigoureuse qui structure le processus de développement en plusieurs phases distinctes, chacune étant accompagnée d'une phase de test correspondante. Ce modèle suit une approche séquentielle, où chaque étape de conception doit être complétée avant de passer à la suivante.

Cette approche intégrée est choisie pour ce projet afin d'avoir un planning strict de réalisation et aussi en même temps apporter des modifications possibles résultants des conseils, remarques et recommandations venant du maître de stage et du directeur de mémoire à la fin de chacune des itérations, spécialement, à la phase de validation de chacune des phases du modèle en V. (8)

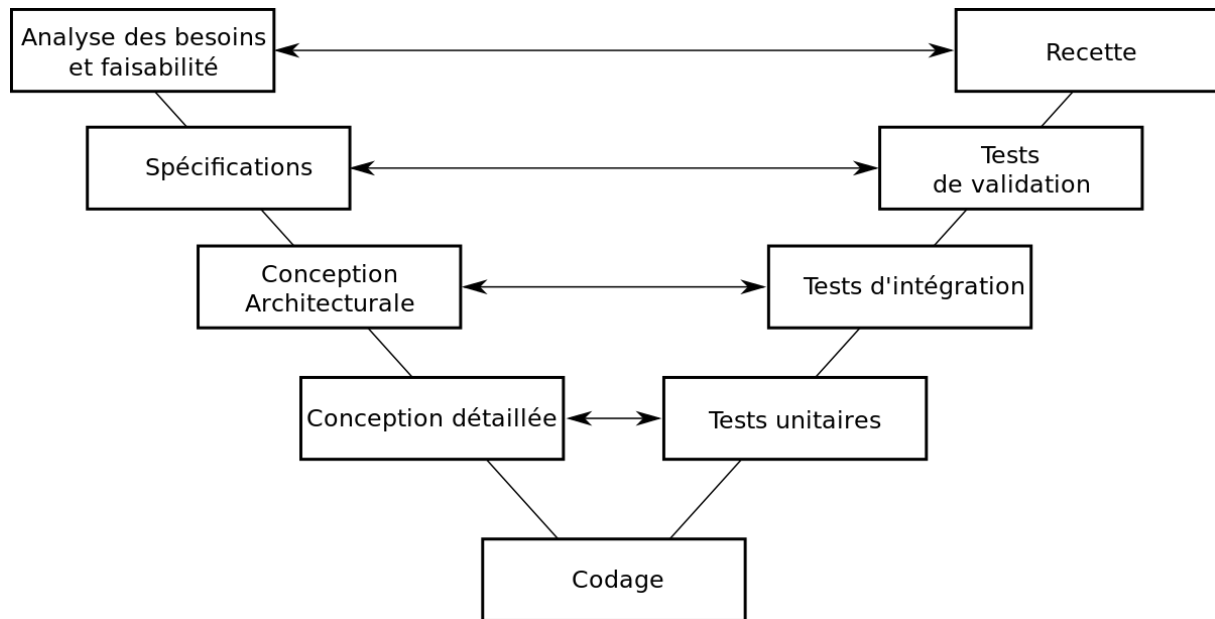


Figure 3 : Schématisation du modèle en V

#### 4.1.2.1 Spécification des besoins utilisateurs

Le processus a commencé par un entretien avec le maître de stage, durant lequel l'idée générale du projet a été exposée. Lors de cette rencontre, les besoins fonctionnels et non fonctionnels du projet ont été explicitement définis, couvrant les attentes en termes de fonctionnalités principales ainsi que les contraintes techniques et opérationnelles. Un document formalisant ces spécifications a été rédigé, récapitulant l'ensemble des besoins, pour s'assurer de la bonne compréhension de l'objectif du projet.

#### 4.1.2.2 Test de recette

Une fois les spécifications des besoins utilisateurs sont définies, le document a été soumis au maître de stage pour l'approbation. La validation a marqué la transition vers l'élaboration d'un cahier des charges, avec la certitude que les exigences des utilisateurs avaient été correctement prises en compte.

#### 4.1.2.3 Spécification des exigences système

Après avoir analysé les besoins utilisateurs, un ensemble d'exigences a été défini, permettant de clarifier les attentes et les fonctionnalités clés du système. Ces exigences ont servi de base pour



l'élaboration d'un cahier des charges formel, adressé à Nova Lead pour officialiser les spécifications techniques, les contraintes fonctionnelles et les performances attendues du projet.

#### **4.1.2.4 Test système**

Le cahier des charges a ensuite été soumis au maître de stage pour validation. Ce dernier a demandé des ajustements, ajoutant certaines fonctionnalités jugées plus prioritaires et en retirant d'autres moins pertinentes. Une fois ces modifications intégrées, la version finale du cahier des charges a été approuvée.

#### **4.1.2.5 Conception globale (système)**

La conception globale se concentre sur la définition de l'architecture du système. L'architecture adoptée est celle en vigueur chez Nova Lead, incluant l'organisation des modules et leurs interactions, selon les pratiques et standards établis par l'entreprise.

#### **4.1.2.6 Test d'intégration**

Le test d'intégration n'a pas été fait pour la conception globale mais après la phase de codage.

#### **4.1.2.7 Conception détaillée**

Lors de la conception détaillée, des diagrammes UML ont été utilisés pour spécifier en profondeur chaque module. Les diagrammes ont inclus des détails sur les interactions entre les modules, les flux de données et les spécificités des fonctionnalités. Une description textuelle a également été réalisée pour fournir des informations supplémentaires sur l'implémentation de chaque fonctionnalité.

#### **4.1.2.8 Test unitaire**

Les diagrammes UML et les descriptions de la conception détaillée ont été présentés au maître de stage et au directeur de mémoire. Le maître de stage a formulé des remarques, tandis que le directeur de mémoire a proposé des améliorations pour une meilleure clarté et précision. Suite à ces retours, certains diagrammes ont été modifiés pour mieux refléter les exigences et optimiser la conception.

#### 4.1.2.9 Réalisation (implémentation)

La phase de codage a débuté avec le développement de l'API, suivi par la création de l'interface. Après avoir développé et intégré ces deux composants, l'ensemble a été soumis au directeur de mémoire.

##### 4.1.2.10 Tests finaux

Le test d'intégration et unitaire ont été fait avec correction des erreurs de logique et les tests finaux sont fait devant le Maître de stage qui a donné son appréciation et encourage ces remarques.

### 4.2 Présentation de l'outils de modélisation

Les outils de modélisation sont essentiels pour visualiser, concevoir et documenter des systèmes et des processus complexes. Ils permettent de créer des diagrammes, des schémas, et des modèles qui aident à la compréhension, à la communication et à l'analyse des systèmes. Draw.io est l'outil de modélisation choisi pour ce projet (9). C'est une plateforme en ligne gratuite et intuitive qui permet de créer une variété de diagrammes et de schémas.



*Figure 4 : Logo de Draw.io*

## 4.3 Etude détaillé de la solution

Un plan détaillé est crucial pour guider efficacement le projet à travers ses différentes phases. Cette partie du projet se base sur le langage UML (Langage de Modélisation Unifié) pour modéliser et structurer les éléments du système de manière visuelle et intégrée. Les diagrammes UML utilisés incluent les diagrammes de cas d'utilisation, les diagrammes de classes, et les diagrammes d'activités. La section suivante présentera en détail ces diagrammes ainsi que la manière dont ils s'intègrent dans le plan global du projet, offrant une vue claire des processus et des livrables associés à chaque phase de développement.

### 4.3.1 Identification des Acteurs

Un acteur ici, représente un utilisateur ou un autre système qui interagit avec ce système de gestion d'assistance en ligne. Selon l'analyse, Trois (3) acteurs sont recensés :

- **Client** : Utilisateur final du système, il peut soumettre de nouveaux tickets d'assistance pour signaler des problèmes ou demander de l'aide, suivre l'état de ses tickets, échanger des messages avec l'agent de support et évaluer la qualité de l'assistance reçue une fois le ticket clôturé.
- **Agent support** : Responsable de la gestion des tickets de support et de la communication avec les clients. Il peut consulter et mettre à jour l'état des tickets, répondre aux messages des clients, et assurer le suivi des interactions. L'agent support peut visualiser les informations des clients associées aux tickets, mais n'a pas accès à la gestion des utilisateurs.
- **Administrateur** : Chargé de la gestion globale du système, incluant la création, la modification, l'archivage des informations concernant les services, les clients et les utilisateurs. L'administrateur supervise l'ensemble du flux de gestion des tickets, peut clôturer des tickets et dispose des droits les plus élevés sur la plateforme.

### 4.3.2 Diagramme de contexte statique

Ce diagramme présente les entités externes qui interagiront avec le système de gestion de Nova Lead.

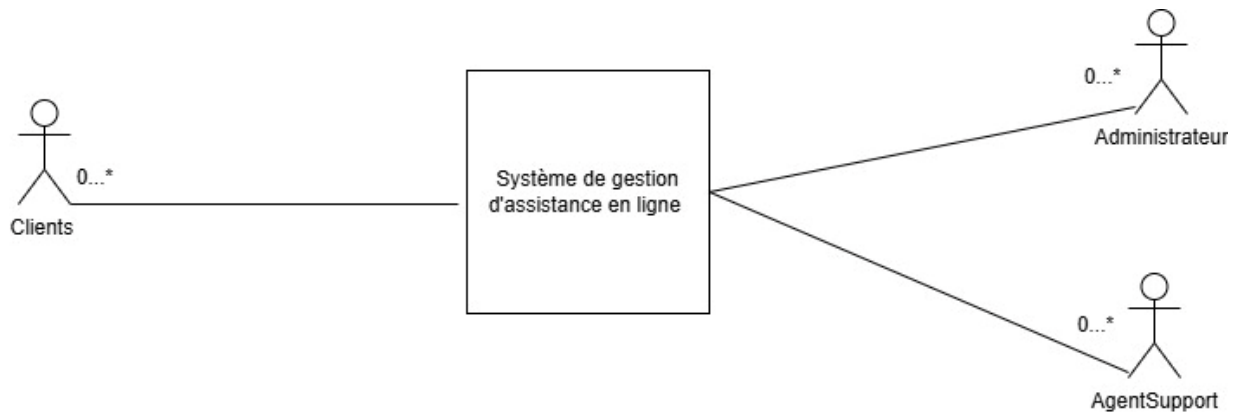


Figure 5 : : Diagramme du contexte statique

### 4.3.3 Diagramme des cas d'utilisation

Le diagramme de cas d'utilisation illustre les interactions entre les acteurs et le système. Il montre les différentes fonctionnalités que le système offre et comment les utilisateurs interagissent avec ces fonctionnalités. Chaque cas d'utilisation représente une fonction ou un service spécifique que le système fournit, tandis que les acteurs sont les entités externes qui utilisent ces services. Ce diagramme aide à comprendre les exigences fonctionnelles du système et les besoins des utilisateurs.

#### 4.3.3.1 Identification des cas d'utilisation par acteur

Le cas d'utilisation représente une fonctionnalité du système (visible de l'extérieur du système) et qui a un intérêt métier pour un acteur donné.

##### a. Administrateur

Les actions suivantes seront réalisées dans le système par un administrateur :

- Ajouter un client dans le système,
- Modifier les informations d'un client dans le système,
- Archivés un Client,
- Envoyer des Messages liés aux tickets des Clients,
- Consulter les Tickets assignés,

- Ajouter un produit dans le système,
- Modifier les informations du produit,
- Archiver un produit,
- Ajouter un type de demande,
- Modifier les informations d'un type de demande,
- Archiver un type de demande
- Ajouter des documents dans le système

#### **b. Agent-Support**

Les actions suivantes seront réalisées dans le système par un Agent-Support :

- Consulter les Tickets assignés,
- Répondre aux messages liés aux tickets des client,
- Ajouter un produit dans le système,
- Modifier les informations du produit,
- Archiver un produit,
- Ajouter un type de demande,
- Modifier les informations d'un type de demande,
- Archiver un type de demande
- Ajouter des documents dans le système

#### **c. Client**

Les actions suivantes seront réalisées dans le système par un Client :

- Soumettre un Ticket d'assistance,
- Envoyer des messages liés à un ticket,
- Modifier les informations d'un ticket,
- Consulter l'historique des Tickets,
- Clôturer un Ticket,
- Evaluer la qualité du support d'un ticket,

### 4.3.3.2 Schématisation

La schématisation est faite par acteur à savoir : l'administrateur, L'agent support et le client

#### a. Administrateur

Le diagramme ci-dessous montre que l'acteur Administrateur peut gérer tous les aspects liés aux clients, aux tickets, aux produits et aux types de demande, avec une relation <<includes>> qui indique qu'il doit obligatoirement s'authentifier pour accéder à ces fonctionnalités.

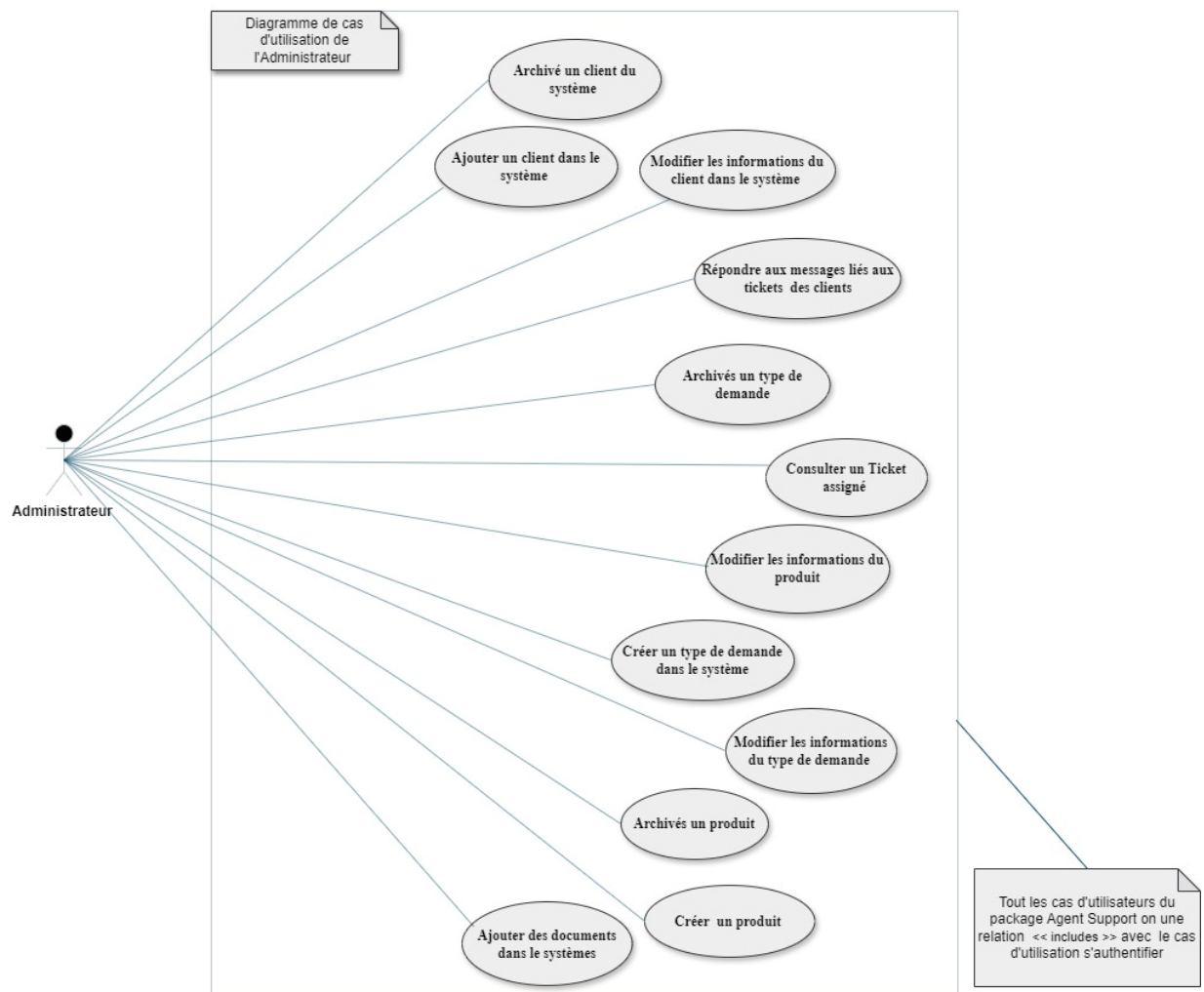


Figure 6 : Diagramme des cas d'utilisation « Administrateur »

## b. Agent support

Le diagramme ci-dessous montre que l'acteur **Agent Support** peut gérer les tickets qui sont assignés, ainsi que créer, modifier et archiver des produits et des types de demande. Toutes ces actions nécessitent une relation <<includes>> avec le cas d'utilisation s'authentifier, garantissant la sécurité et la traçabilité des actions réalisées par cet acteur.

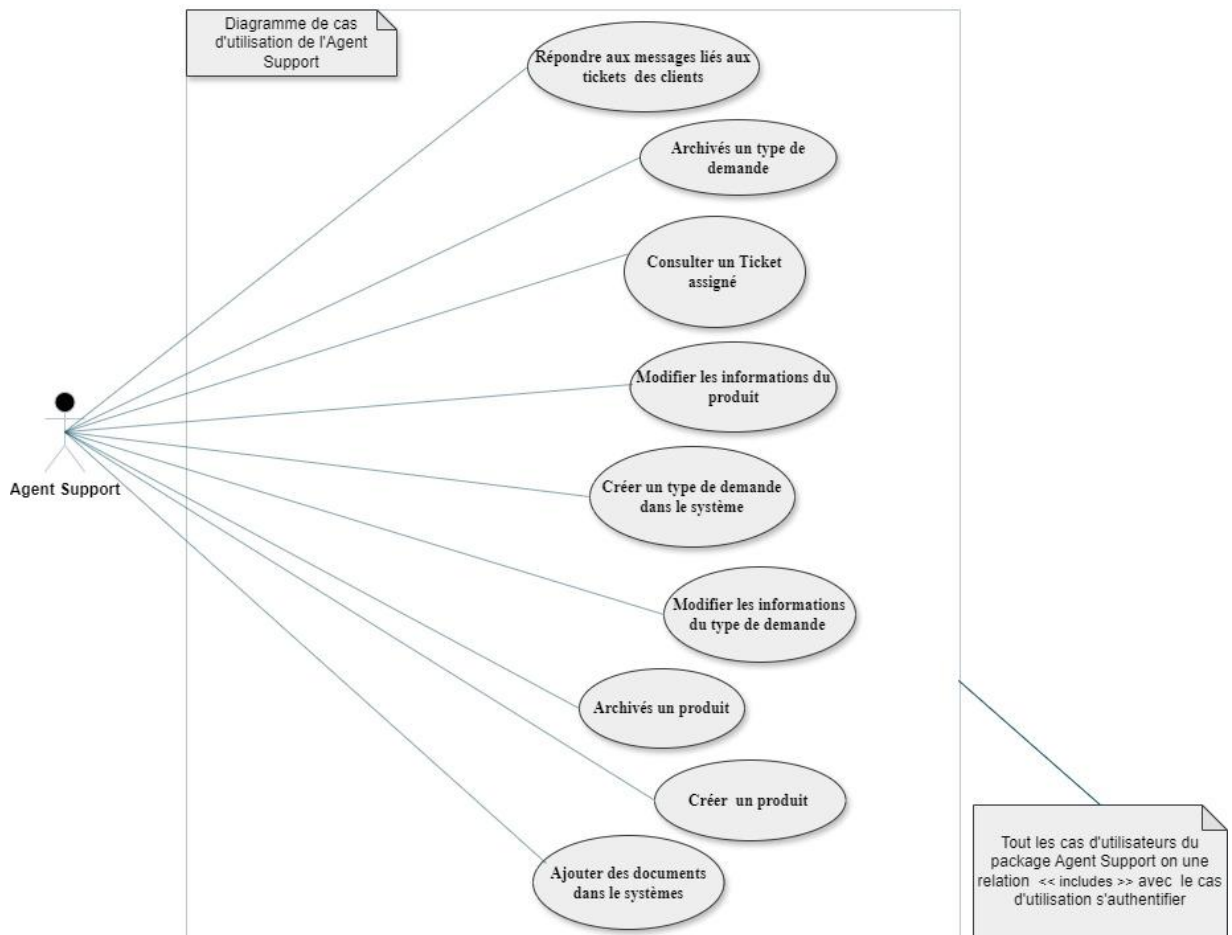


Figure 7 : Diagramme des cas d'utilisation « Agent Support »

### c. Client

Le diagramme ci-dessous illustre les cas d'utilisation liés au processus de gestion des tickets d'assistance. Il met en évidence les différentes interactions possibles, telles que la soumission et la modification de tickets, ainsi que d'autres actions essentielles, comme l'envoi de messages, la consultation de l'historique des tickets, l'évaluation de la qualité du support, et la clôture des tickets après résolution. Il précise également la nécessité pour chaque client de s'authentifier avant d'accéder à ces fonctionnalités.

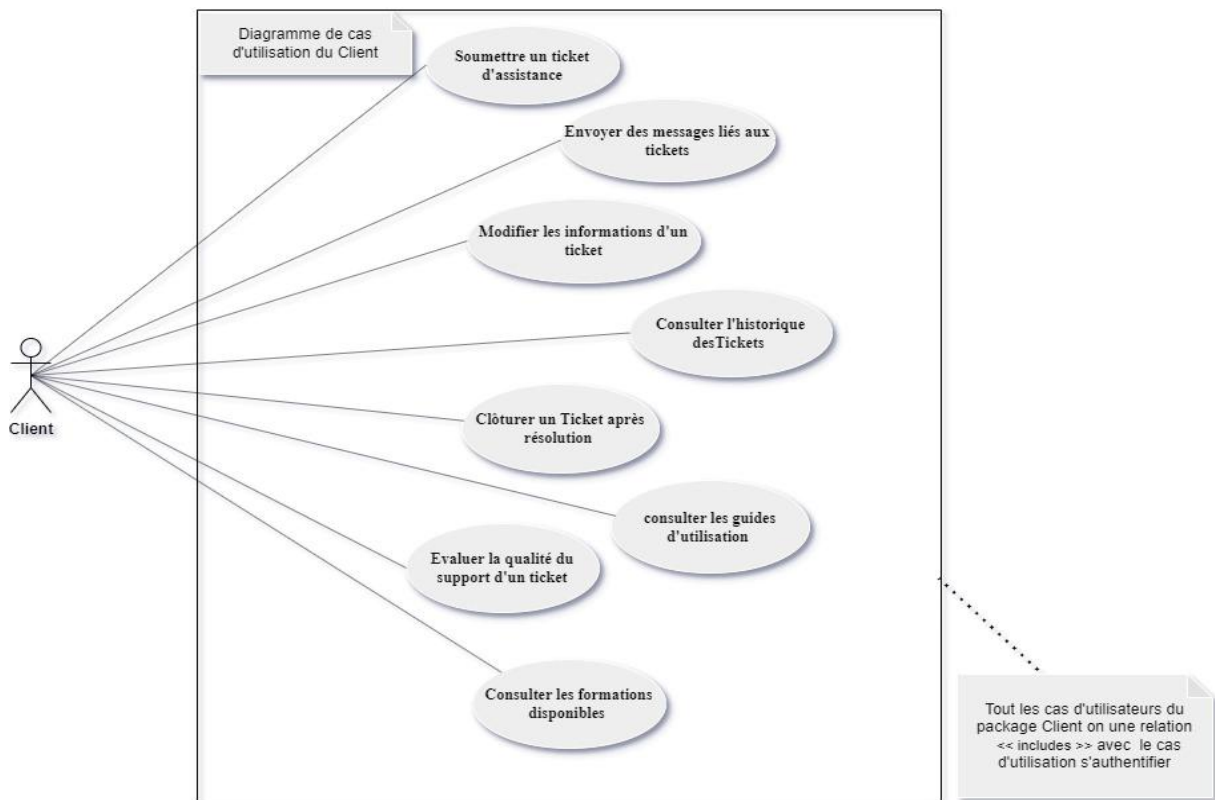


Figure 8 : Diagramme des cas d'utilisation « Client »



### 4.3.4 Description textuelle des cas d'utilisation

Elle fournit une description détaillée des cas d'utilisation. Dans ce document, seuls certains cas d'utilisation sont détaillés, tels que : enregistrer un client, enregistrer un produit, enregistrer un type de demande, créer un ticket.

#### 4.3.4.1 Enregistrer un client

Le tableau ci-dessous décrit l'interaction entre le système et l'administrateur lors de l'enregistrement d'un client.

Tableau 4 : Description textuelle du cas d'utilisation " enregistrer un client "

Titre	Enregistrer un client
Résumé	Ce cas d'utilisation permet à l'administrateur d'ajouter un nouveau client
Acteur	Administrateur
Précondition	L'authentification de l'administrateur
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur demande la page d'ajout de client.</li> <li>2. Le système affiche la page d'ajout de client.</li> <li>3. L'administrateur remplit le formulaire d'ajout de client.</li> <li>4. L'administrateur envoie le formulaire.</li> <li>5. Le système vérifie les données du formulaire.</li> <li>6. Le système affiche un message de succès.</li> </ol>
Scénario alternatif	<p>A1 : les données des champs sont invalides.</p> <p>L'enchaînement déclenche à l'étape 5</p> <p>6. Le système renvoie un message d'erreur.</p> <p>Le scénario nominal reprend à l'étape 3</p>
Scénario d'erreur	-

#### 4.3.4.2 Enregistrer un Produit

Le tableau ci-dessous décrit l'interaction de l'administrateur, agent-support et le système lors de l'enregistrement des différents produits ou solutions qu'offre Nova Lead.

Tableau 5 : Description textuelle du cas d'utilisation « enregistrer un produit »

Titre	Enregistrer un Produit
Résumé	Ce cas d'utilisation permet à l'utilisateur d'ajouter un nouveau Produit
Acteur	Administrateur, Agent Support
Précondition	L'authentification de l'administrateur et l'agent-support
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'administrateur, Agent demande la page d'ajout de Produit.</li> <li>2. Le système affiche la page d'ajout de produit.</li> <li>3. L'administrateur, Agent remplit le formulaire d'ajout de produit.</li> <li>4. L'administrateur, Agent envoie le formulaire.</li> <li>5. Le système vérifie les données du formulaire.</li> <li>6. Le système affiche un message de succès.</li> </ol>
Scénario alternatif	<p>A1 : les données des champs sont invalides.</p> <p>L'enchaînement déclenche à l'étape 5</p> <p>7. Le système renvoie un message d'erreur.</p> <p>Le scénario nominal reprend à l'étape 3</p>
Scénario d'erreur	-

#### 4.3.4.3 Enregistrer un type de demande

Le tableau ci-dessous décrit l'interaction entre l'Administrateur ou Agent Support et le système lors de l'enregistrement d'un type de demande.

Tableau 6 Description textuelle du cas d'utilisation « enregistrer un nouveau type de demande »

Titre	Enregistrer un nouveau type de demande
Résumé	Ce cas d'utilisation permet à l'administrateur ou l'agent support d'enregistrer un nouveau type de demande.
Acteur	Administration ou Agent Support
Précondition	L'authentification de l'administrateur ou l'agent support
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'Administrateur ou l'Agent support demande une page d'ajout d'un nouveau type de demande</li> <li>2. Le système affiche la page d'enregistrement de type de demande.</li> <li>3. L'administrateur ou l'agent support remplit le formulaire d'enregistrement d'un type de demande.</li> <li>4. Administrateur ou l'Agent support soumet le formulaire.</li> <li>5. Le système vérifie les données du formulaire.</li> <li>6. Le système affiche un message de succès d'enregistrement d'un nouveau type de demande.</li> </ol>
Scénario alternatif	<p>A1 : les données des champs sont invalides.</p> <p>L'enchaînement déclenche à l'étape 5</p> <p>6. Le système renvoie un message d'erreur.</p> <p>Le scénario nominal reprend à l'étape 3</p>
Scénario d'erreur	-

#### 4.3.4.4 Enregistrer un Ticket

Le tableau ci-dessous décrit l'interaction entre le système et le Client lors de l'enregistrement d'un Ticket

Tableau 7 : Description textuelle du cas d'utilisation « enregistrer un Ticket »

Titre	Enregistrer un nouveau Ticket
Résumé	Ce cas d'utilisation permet à l'utilisateur d'enregistrer un nouveau ticket
Acteur	Client
Précondition	<ol style="list-style-type: none"> <li>1. L'authentification du Client</li> <li>2. L'existence d'un produit</li> <li>3. L'existence d'un type de demande</li> <li>4. L'existence de l'urgence</li> </ol>
Scénario nominal	<ol style="list-style-type: none"> <li>5. Le client demande la page d'ajout de ticket.</li> <li>6. Le système affiche un formulaire d'ajout de ticket</li> <li>7. Le client clique sur les barres de sélection pour choisir le produit, l'urgence et type de demande</li> <li>8. Le client renseigne les informations manquantes du formulaire Telle que la description, joindre un fichier, ...</li> <li>9. Le client envoie le formulaire.</li> <li>10. Le système vérifie les données du formulaire.</li> <li>11. Le système affiche un message de succès d'ajout de ticket</li> </ol>
Scénario alternatif	-
Scénario d'erreur	-

#### 4.3.4.5 Enregistrer une Note

Le tableau ci-dessous décrit l'interaction entre le système et le client lors de l'enregistrement d'une note.

Tableau 8 : Description textuelle du cas d'utilisation « enregistrer une note »

Titre	Enregistrer une Note
Résumé	Ce cas d'utilisation permet au client d'attribuer une note à un ticket
Acteur	Client
Précondition	L'authentification du Client
Scénario nominal	<ol style="list-style-type: none"> <li>1. Le client demande la page d'ajout de note.</li> <li>2. Le système affiche la page d'ajout de note.</li> <li>3. Le Client remplit le formulaire d'ajout de note.</li> <li>4. Le client envoie le formulaire.</li> <li>5. Le système vérifie les données du formulaire.</li> <li>6. Le système affiche un message de succès.</li> </ol>
Scénario alternatif	<p>A1 : les données des champs sont invalides.</p> <p>L'enchaînement déclenche à l'étape 5</p> <p>6. Le système renvoie un message d'erreur.</p> <p>Le scénario nominal reprend à l'étape 3</p>
Scénario d'erreur	-

#### 4.3.5 Diagramme d'activités

Le diagramme d'activités représente les flux de travail ou les processus liés aux interactions du client avec le système. Il illustre les différentes étapes (activités) effectuées par le client, les décisions prises à chaque étape, ainsi que les transitions entre ces activités. Ce diagramme est particulièrement utile pour décrire les processus associés à l'utilisation du système par le client, en fournissant une vue d'ensemble des interactions et des flux de données spécifiques à ses cas d'utilisation. Voici présenté le comportement du système lors du déroulement des cas d'utilisation du client, de l'administrateur et l'agent support tels que l'enregistrement d'un ticket, l'évaluation du support, enregistrement d'un client, ...

### 4.3.5.1 Enregistrer un client

Le diagramme ci-dessous est une schématisation de la description de l'interaction entre le système et l'administrateur lors de l'enregistrement un client.

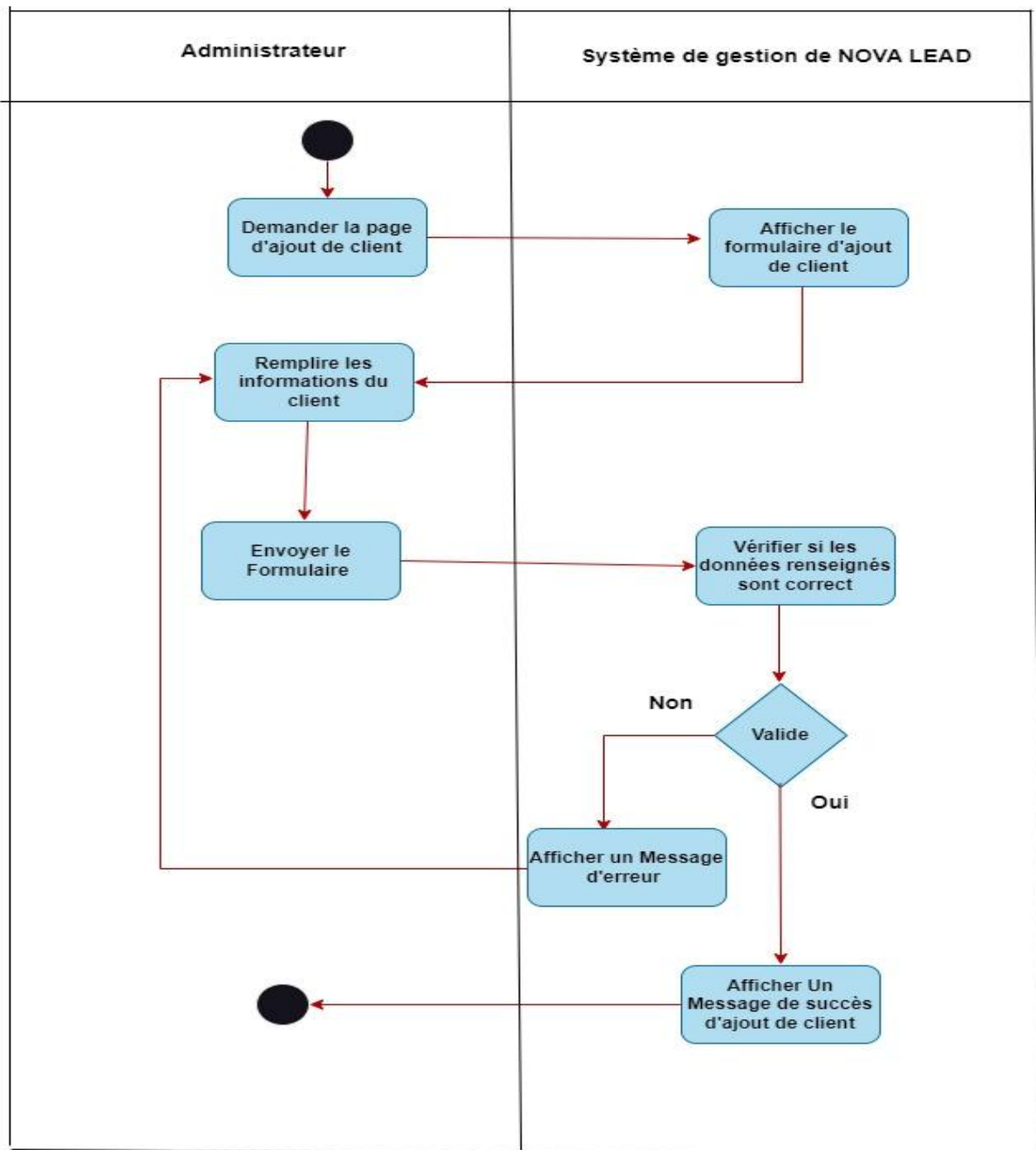


Figure 9 : Diagramme d'activité du cas d'utilisation « enregistrer un client »

### 4.3.5.2 Enregistrer un ticket

Le diagramme ci-dessous est une schématisation de la description de l'interaction entre le système et le client lors de l'enregistrement d'un ticket

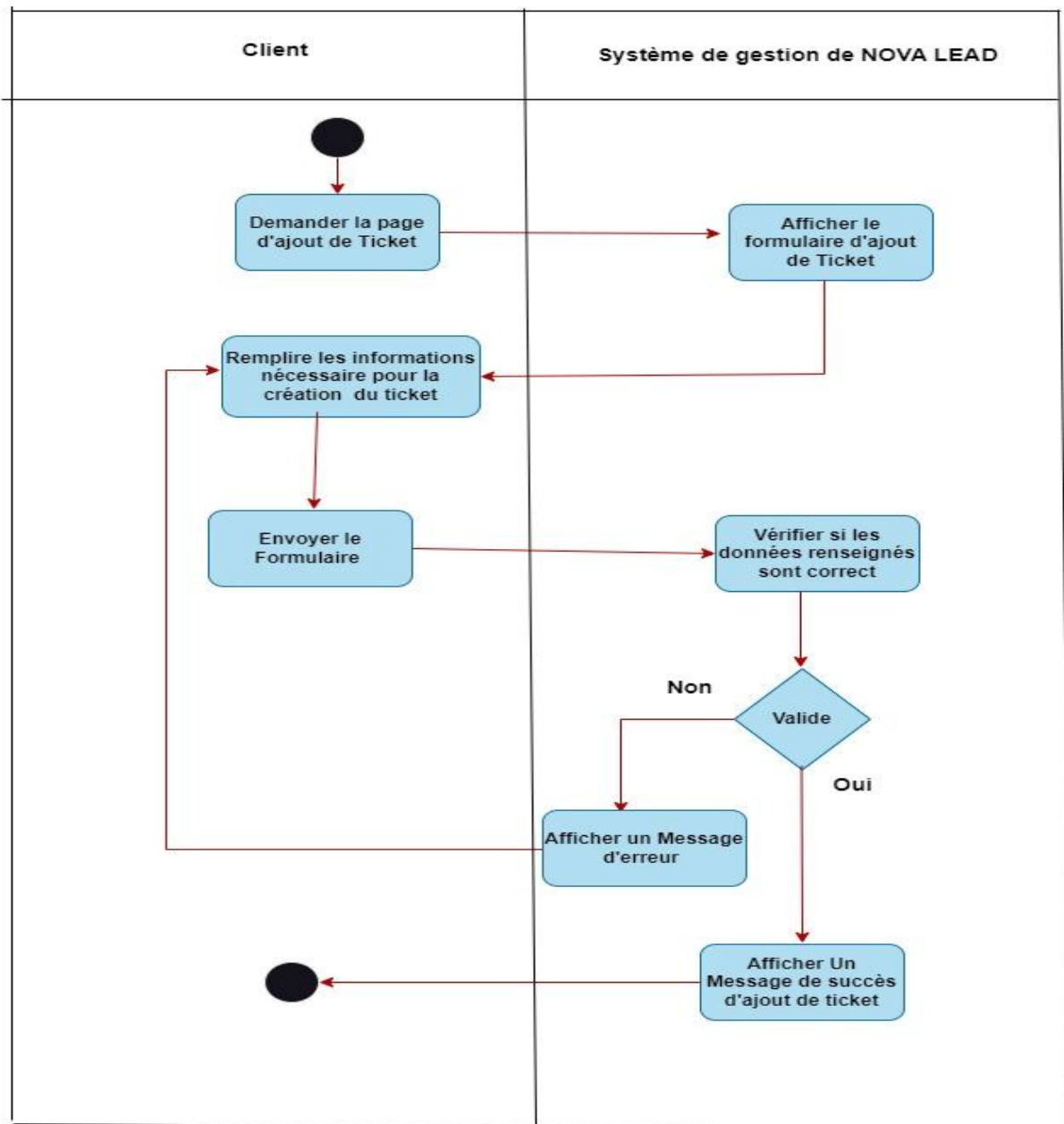


Figure 10 : Diagramme d'activité du cas d'utilisation « enregistrer un ticket »

### 4.3.5.3 Ajouter une note

Le diagramme ci-dessous illustre les interactions entre système et le client lors de l'attribution d'une note.

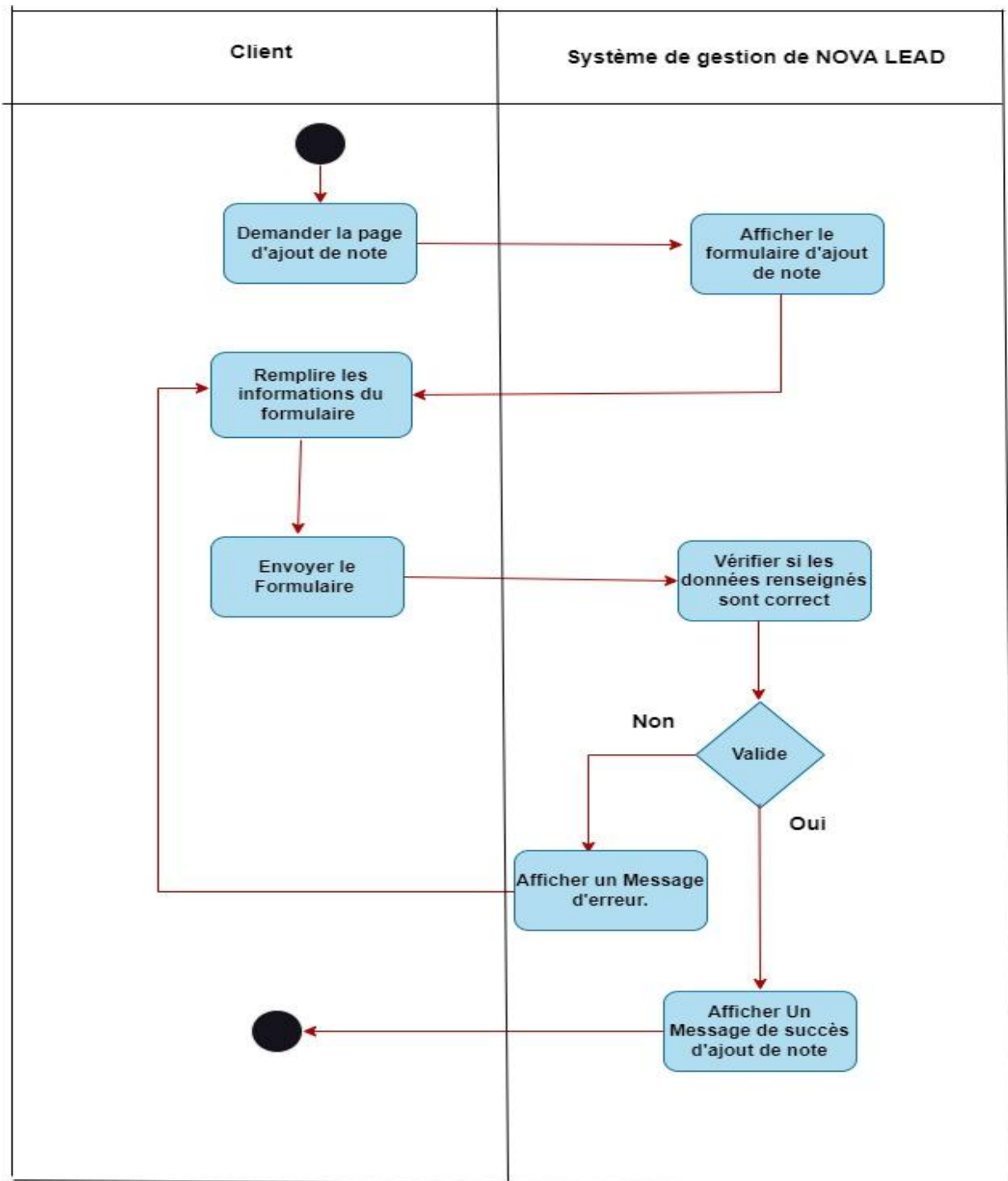


Figure 11 : Diagramme d'activité du cas d'utilisation « ajouter une note »



### 4.3.6 Diagramme des classes

Le diagramme des classes décrit la structure statique du système en montrant les différentes classes d'objets, leurs attributs, leurs méthodes et les relations entre elles. Il représente les entités du système et les associations entre elles, telles que l'héritage, l'association et la composition. Ce diagramme est essentiel pour concevoir la structure du système et pour comprendre comment les différentes parties du système interagiront.

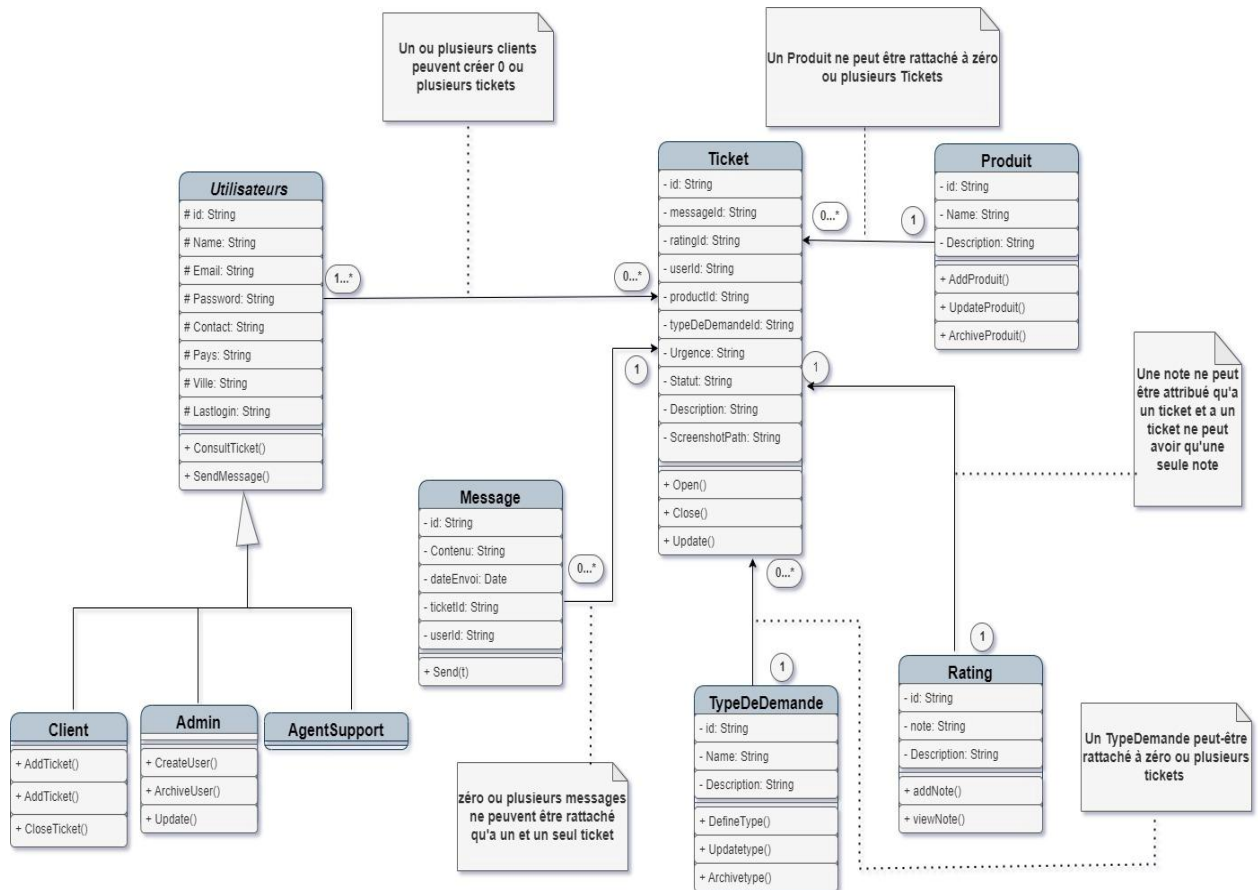


Figure 12 : Diagramme des classes du système de gestion de Nova Lead

## CHAPITRE 4 : RÉALISATION ET MISE EN ŒUVRE

Cette section présente les outils et moyens de mise en œuvre du système de gestion d'assistance de Nova Lead et la présentation du système obtenu.

### 7.1 Mise en œuvre

#### 7.1.1 Choix de matériel

Le choix du matériel s'est focalisé sur des équipements performants et compatibles avec les exigences du projet, garantissant ainsi une bonne évolutivité et une mise en œuvre fluide et rapide.

##### 7.1.1.1 Ordinateurs

Le laptop décrit dans le tableau ci-dessous, a été l'outil de création des diagrammes UML et de développement des fonctionnalités du système de gestion d'assistance de Nova Lead. Ses performances et sa portabilité en font le choix idéal pour travailler partout, sans perdre en efficacité.

Tableau 9 : Caractéristiques du matériel de développement

Élément	Valeur
Processeur	8th Gen Intel(R) Core (TM) i7-8665U CPU @ 1.90GHz 2.11 GHz, 1690 MHz, 4 cœurs, 8 processeurs logiques
Mémoire RAM	16,0 Go
Type de RAM	DDR4
Mémoire ROM	500 Go
Type de ROM	SSD
Fabricant	Dell Inc.
Modèle	Latitude 5400
Type du PC	PC à base de x64

### **7.1.2 Choix de logiciel**

Le choix du logiciel est une étape clé du développement, car il impacte directement la productivité et la qualité du projet. Plusieurs critères, comme la flexibilité et la facilité d'utilisation, ont été pris en compte pour choisir un outil capable de répondre aux besoins techniques et d'accélérer la mise en œuvre.

#### **7.1.2.1 Environnement de développement**

Visual Studio Code (VS Code) est un éditeur de code source développé par Microsoft. Conçu pour être léger et performant, il prend en charge un large éventail de langages de programmation et de technologies grâce à des extensions, ce qui le rend très flexible. Pour la réalisation du système de gestion d'assistance de Nova Lead, VS Code a été choisi pour le développement de l'API ainsi que l'implémentation des interface UI, en raison de sa légèreté, de ses performances, et de sa grande flexibilité. Il est rapide, même sur des machines aux ressources limitées, et propose une large gamme d'extensions adaptées aux besoins spécifiques du développement, telles que le support pour Node.js et la gestion de bases de données.

#### **7.1.2.2 Système d'exploitation**

La machine choisie pour la réalisation du projet, décrite plus haut, est dotée de Windows 11 professionnel, un produit développé par Microsoft et conçu pour les ordinateurs personnels comme celui choisi pour le ce projet. Son intuitivité et sa facilité d'utilisation ainsi que sa rapidité prouve l'évidence de son choix pour ce projet.

#### **7.1.2.3 Conception de l'interface utilisateur**

Figma a été un outil de conception des interfaces utilisateur (UI) de l'application Nova Lead. Grâce à son environnement de conception collaboratif et basé sur le cloud, Figma a permis de créer des maquettes interactives et de visualiser rapidement les différentes composantes de l'interface facilitant ainsi l'implémentation des interfaces UI en Vue JS.

#### 7.1.2.4 Technologies de programmation

Le choix du langage de programmation est une étape cruciale dans la mise en place de ce système informatique. Pour ce projet, plusieurs critères ont été pris en compte afin de sélectionner le langage le plus adapté aux besoins de l'entreprise Nova Lead. Les raisons de l'utilisation des technologies sont :

- **Performance et efficacité** : Les technologies choisies doivent garantir une exécution rapide et fluide, notamment pour la gestion des tickets, la communication en temps réel et le traitement des messages entre les acteurs.
- **Évolutivité et maintenabilité** : le système étant amené à évoluer au fil du temps, il est crucial de choisir un langage qui facilite la modification du code et l'ajout de nouvelles fonctionnalités, tout en garantissant la stabilité du système. De plus, l'entreprise d'accueil étant spécialisée dans ce langage pour le développement, cela offre un support continu tout au long du projet et assure une évolution pérenne du système. Ainsi, la maintenabilité ne sera pas un obstacle au bon déroulement et à l'adaptabilité future du système.
- **Communauté et support** : un langage avec une large communauté offre plus de ressources, de bibliothèques et de support, facilitant ainsi le développement et la résolution des problèmes techniques.
- **Interopérabilité** : le système devant interagir avec d'autres technologies comme les bases de données ou les API, le langage choisi doit garantir une bonne compatibilité et communication avec ces technologies.

##### a. Langages de Programmation

Les langages utilisés lors de la réalisation du projet :

- **JavaScript** : Utilisé côté serveur avec Node.js et côté client avec Vue.js, JavaScript est un langage flexible et puissant, idéal pour le développement des applications web interactives et performantes.

##### b. Framework

Les frameworks utilisés lors de la réalisation du projet :

- **Node.js** : environnement d'exécution JavaScript côté serveur, permettant une gestion asynchrone des requêtes et une grande scalabilité pour les applications web.

- **Vue.js** : Framework JavaScript côté client, permettant de créer des interfaces utilisateur interactives, modulaires et performantes pour les applications web.

De plus, le besoin d'intégrer une API pour interagir avec divers services et sources de données a été pris en compte. Vue.js offre une excellente compatibilité avec les API backend, et l'entreprise d'accueil étant spécialisée dans ce domaine assure une expertise solide dans la mise en œuvre des solutions Vue.js.

#### 7.1.2.5 Langage de requête

REST API a été choisi comme approche pour l'API en raison de sa simplicité et de sa compatibilité avec divers clients front-end, tels que Vue.js. REST permet de structurer les requêtes de manière standardisée, facilitant ainsi la récupération des données nécessaires et leur gestion. Il offre également une bonne extensibilité, permettant d'ajouter de nouvelles fonctionnalités sans affecter les requêtes existantes.

#### 7.1.2.6 Base de données

**MongoDB** est une base de données NoSQL orientée documents, idéale pour gérer des données structurées et semi-structurées sous forme de documents JSON (10). Elle a été choisie pour sa flexibilité dans la manipulation des données et pour sa capacité à évoluer avec les besoins croissants du système. MongoDB permet de gérer efficacement les relations entre les différentes entités du système, telles que les utilisateurs, les tickets, et les produits, grâce à son support natif des relations imbriquées et des références entre documents.

MongoDB utilise le **MongoDB Query Language (MQL)**, qui offre des capacités avancées pour effectuer des requêtes complexes, manipuler les données et gérer les relations tout en assurant des performances optimales pour les applications web comme celle de Nova Lead.

MongoDB utilise un le **Object-Document Mapping (ODM)** pour le mappage des données au format BSON. Dans notre cas c'est L'ODM Mongoose qui a été utilisé à cause de sa popularité et sa complexité.

#### 7.1.2.7 Architecture logicielle de l'application

Notre application d'assistance client repose sur une structure modulaire suivant le modèle **MVC (Modèle-Vue-Contrôleur)**, une approche largement utilisée pour organiser les applications

logicielles modernes. L'architecture Modèle/Vue/Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface. L'organisation globale d'une interface graphique est souvent délicate. Bien que la façon MVC d'organiser une interface ne soit pas la solution miracle, elle fournit souvent une première approche qui peut ensuite être adaptée. Elle offre aussi un cadre pour structurer une application. Le modèle MVC impose donc une séparation totale entre le traitement, l'interface et la communication entre ces deux parties. Cela permet d'avoir non seulement des objets réutilisables pour d'autres applications, mais aussi de pouvoir faire évoluer aisément son programme. Ainsi, si l'on souhaite modifier sa base de données il suffit de revoir son "model" et cela est valable pour le cas où l'on souhaite changer d'interface. Les 3 parties du model MVC sont réellement autonomes. Aucune d'elles ne s'occupent du fonctionnement de l'autre. Dans l'architecture MVC, les rôles des trois entités sont les suivants :

- Modèle : données (accès et mise à jour)
- Vue : interface utilisateur (entrées et sorties)
- Contrôleur : gestion des événements et synchronisation

Notre application est divisée en deux parties principales :

- **Backend** développe avec Node.js (Express.js) pour la gestion des données et la logique métiers
- **Le frontend** développé avec **Vue.js**, qui se charge de l'interface utilisateur (UI) et de la présentation des données.

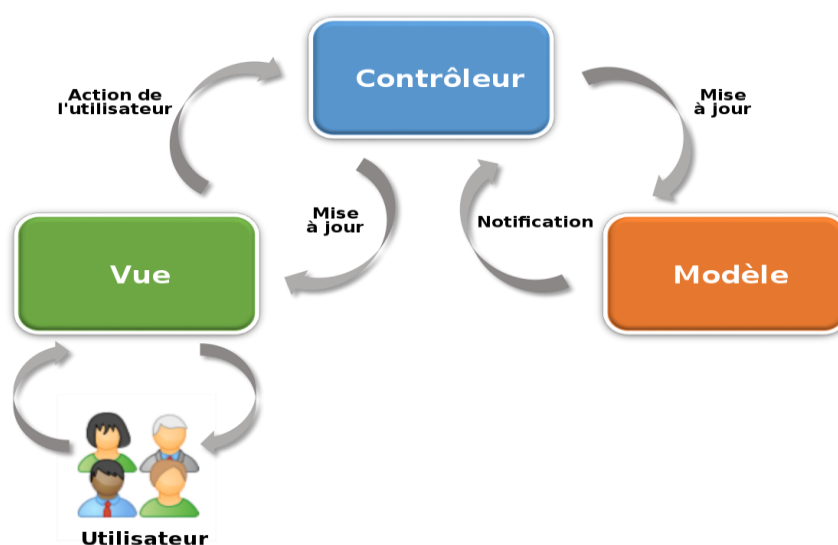


Figure 13 : modèle MVC

### 7.1.2.8 Design Pattern de Node.js

Node.js repose principalement sur une architecture orientée événements, combinée à un modèle non-bloquant, ce qui permet de créer des applications hautement performantes et scalables. L'élément central de Node.js est la boucle d'événements (Event Loop), qui gère de manière asynchrone les tâches d'entrée/sortie(I/O) sans bloquer le thread principal. Ce modèle permet d'exécuter plusieurs opérations simultanément, comme des requêtes http ou des accès à une base de données, sans attendre la fin d'une tâche avant de passer à la suivante. Par ailleurs, Node.js utilise des modules (via CommonJS ou ES Modules) pour structurer le code, favorisant la réutilisation et la gestion des dépendances. Le pattern Singleton est également utilisé dans la gestion des modules, garantissant qu'une seule instance d'un module soit partagée à travers toute l'application. De plus, des patterns comme l'Observer et le Middleware sont intégrés dans des composants comme l'EventEmitter (pour écouter des événements) et les middlewares d'Express.js (pour traiter les requêtes HTTP de manière séquencée). Enfin, le modèle asynchrone, avec l'utilisation de callbacks, Promises et `async/await`, permet d'éviter les blocages et d'améliorer la lisibilité et la maintenabilité du code.

### 7.1.2.9 Système de gestion et d'hébergement du code

Git est un système de contrôle de version distribué qui permet de suivre et de gérer les modifications apportées au code source de manière efficace (11). Pour le développement du système de gestion d'assistance en ligne de Nova Lead, Git a été utilisé pour assurer un suivi rigoureux de l'évolution du code, facilitant ainsi la gestion des versions et des contributions. En parallèle, GitHub a été utilisé comme plateforme d'hébergement en ligne, permettant de centraliser le code source, d'en assurer la sauvegarde du projet.

### 7.1.3 Sécurité de l'application

La sécurité des données est une priorité essentielle dans le développement d'une plateforme d'assistance en ligne. Assurer la protection des données contre les accès non autorisés, les pertes et les altérations est crucial pour maintenir la confidentialité, l'intégrité et la disponibilité des informations.

### 7.1.3.1 Authentification des utilisateurs

La plateforme d'assistance en ligne de Nova Lead dispose d'une page de connexion, et l'authentification est requise pour toutes les actions au sein du système. Lors de la connexion, un token JWT est généré à l'aide du package JWT en Node.js. Ce token assure que chaque requête est authentifiée, garantissant ainsi une sécurité renforcée et un accès contrôlé aux fonctionnalités et données du système.

### 7.1.3.2 Sécurité des données

La sécurité des données est assurée par les mesures suivantes :

- **Sécurisation de la base de données**

La base de données MongoDB est protégée à l'aide de contrôles d'accès stricts, de connexions sécurisées via TLS/SSL et d'une configuration rigoureuse pour empêcher les intrusions.

- **Chiffrement des mots de passe**

Les mots de passe des utilisateurs sont chiffrés avec l'algorithme bcrypt avant d'être stockés dans la base de données. Cela garantit que même en cas d'accès illégal, les mots de passe restent illisibles et protégés.

### 7.1.4 Prévisions financières

Le coût prévisionnel d'un projet représente l'estimation des dépenses nécessaires à sa mise en œuvre. Il englobe les coûts directs et indirects liés aux ressources humaines, matérielles, technologiques, et aux autres services requis pour le bon déroulement du projet. Cette estimation permet de planifier les ressources financières, de définir les budgets et de prévoir les risques financiers. Elle est essentielle pour assurer la viabilité du projet et éviter les imprévus qui pourraient compromettre son succès.

Dans le cadre du projet de gestion de NOVA LEAD, le calcul du coût prévisionnel repose sur plusieurs facteurs clés, notamment : Coût des ressources humaines, Coût des infrastructures. La prévision financière s'élève à 1 720 500 FCFA, détaillée comme suit.



Tableau 10 : Coût financier du projet

	Coût prévisionnel
Internet	25 000 FCFA
Ressources humaines	1 500 000 FCFA
Serveur pour l'hébergement	375 500 FCFA
Maintenance du PC et augmentation de RAM	20 000 FCFA
<b>Coût total</b>	<b>1 720 500 FCFA</b>

## 7.2 Présentation de l'application

Il est désormais temps de présenter le produit obtenu à l'issue de ce projet. L'application développée pour Nova Lead se veut une solution complète, conçue pour répondre aux besoins spécifiques de l'entreprise en matière d'assistance support en ligne

### 7.2.1 Présentation

Cette partie présente brièvement la plateforme d'assistance de Nova Lead. Il présente trois espaces utilisateurs : espace administrateur, espace Agent Support et l'espace client.

### 7.2.1.1 Structure de l'espace administrateur

La figure suivante montre la structure de l'espace administrateur.

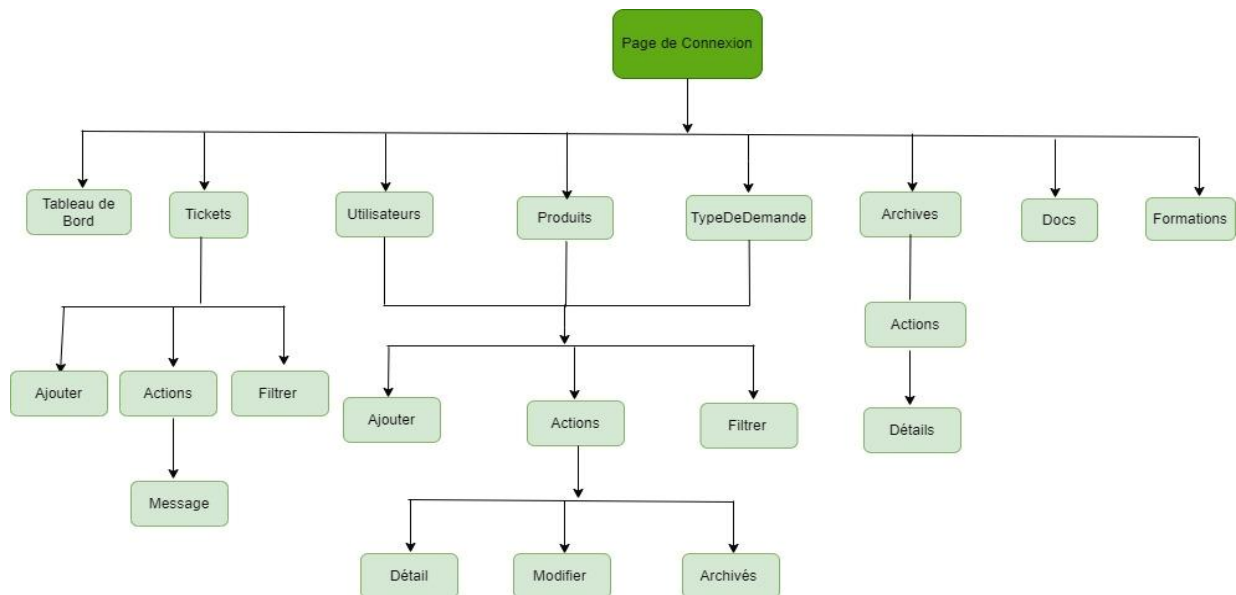


Figure 14 : Structure de l'application – espace Administrateur

### 7.2.1.2 Structure de l'espace d'Agent Support

La figure suivante montre la structure de l'espace Agent Support

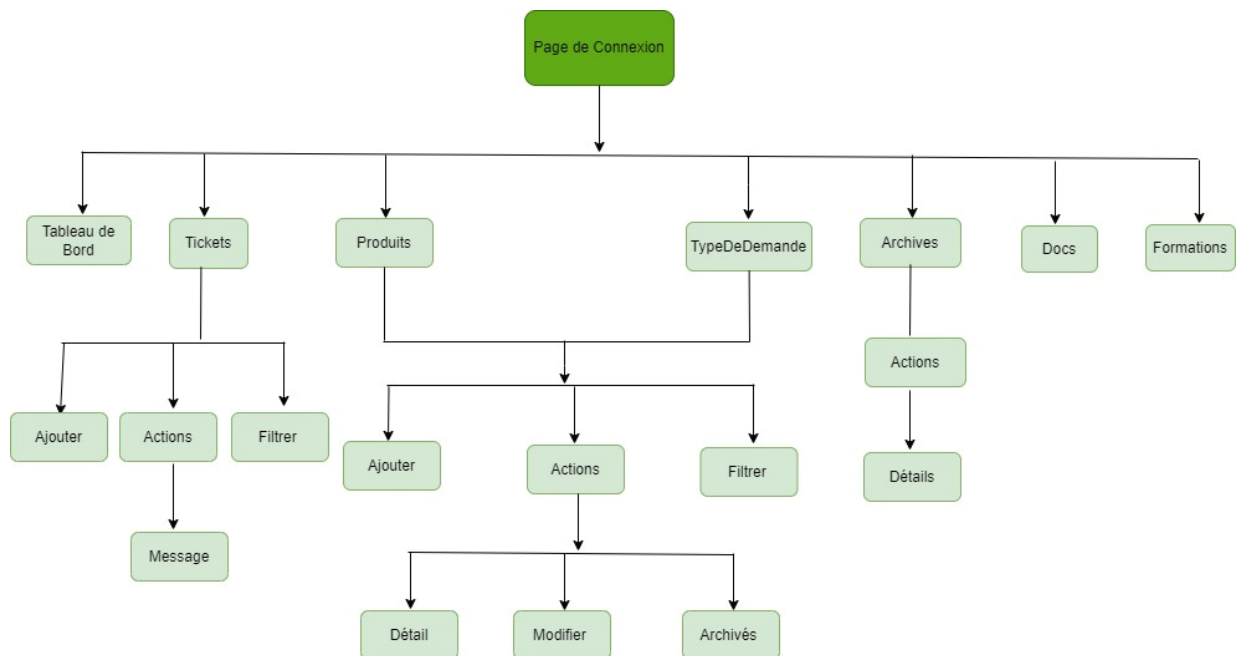


Figure 15 : Structure de l'application – espace d'Agent Support

### 7.2.1.3 Structure de l'espace Client

La figure suivante montre la structure de l'espace Client

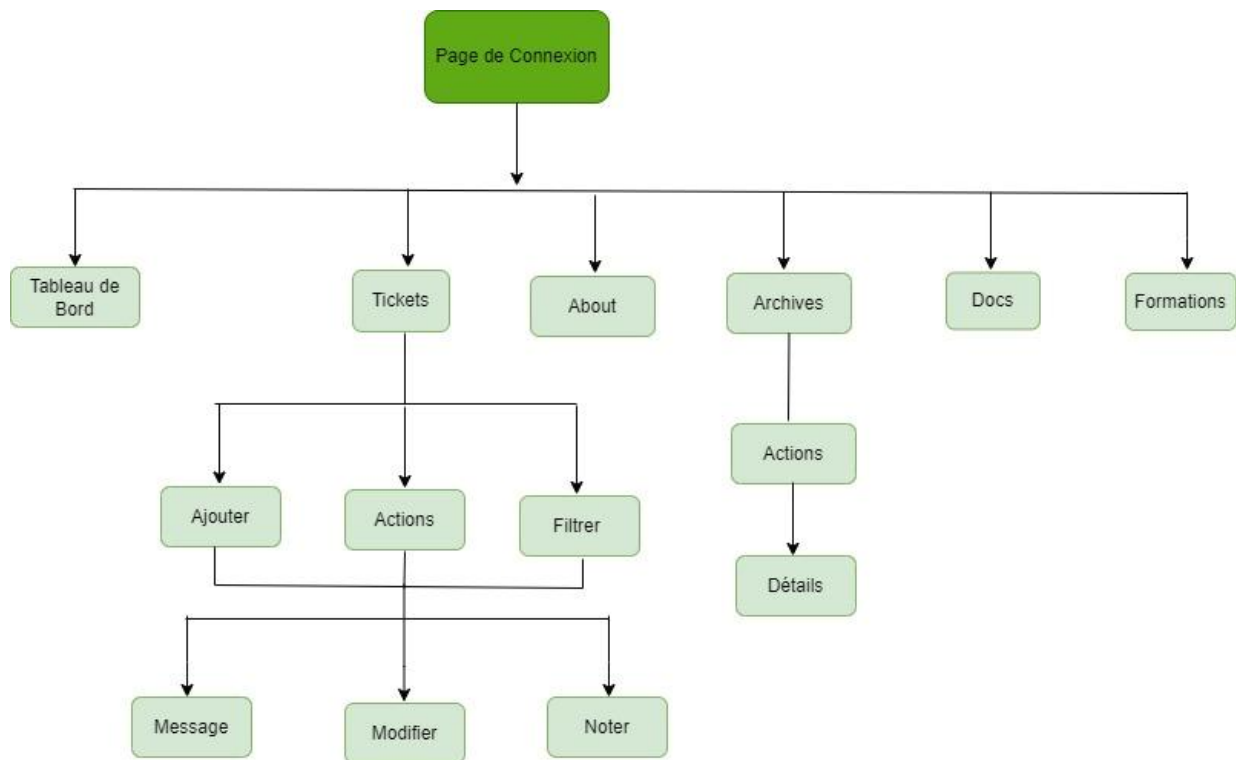


Figure 16 : Structure de l'application – espace Client

## 7.2.2 Architecture de l'application

L'Architecture en Couches est un modèle qui organise les applications en plusieurs couches distinctes, chacune ayant une responsabilité spécifique :

- **Couche de présentation**, interface utilisateur de la plateforme de gestion des tickets d'assistance de Nova Lead. Elle comprend les tableaux de bord pour les clients et administrateurs, permettant respectivement la création et la gestion des tickets.
- **Couche de logique métier**, Elle traite ou gère des règles de gestion liées à la plateforme, comme la création des tickets, la validation des données, la gestion de rôles (client, administrateur ou agent) et le traitement des actions sur les tickets.
- **Couche d'accès aux données**, Elle permet la gestion des interactions avec la base de données. Cette architecture a été choisie pour sa modularité, sa facilité de maintenance, et sa testabilité, permettant une meilleure organisation et évolutivité du système, rendant la gestion des tickets d'assistance plus efficace et structuré.

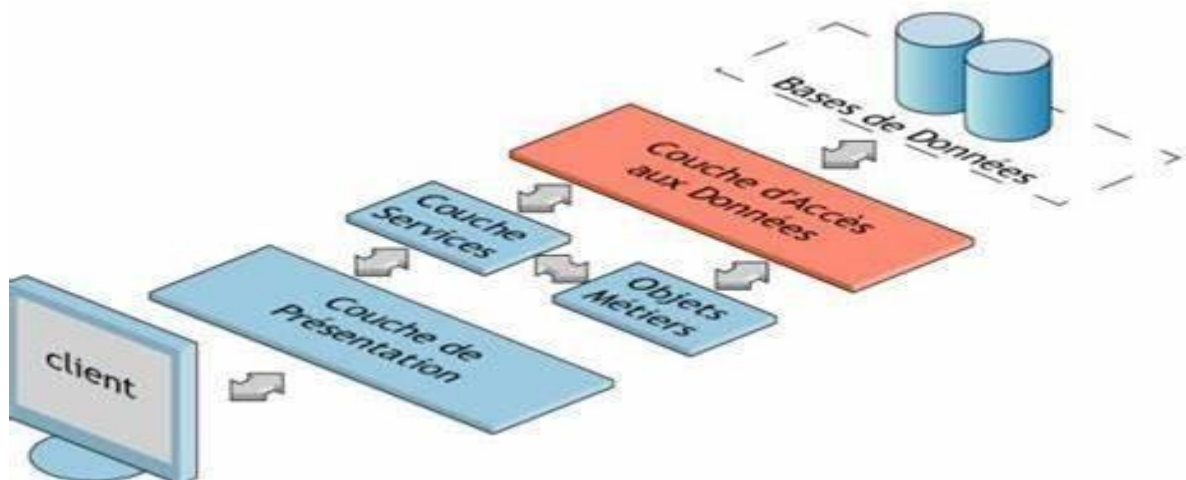


Figure 17 : Architecture de l'application

## 7.2.3 Quelques masques de saisies de la plateforme d'assistance client de Nova Lead

Cette section présente les principaux masques de saisie de l'application développée pour Nova Lead. Ces interfaces permettent aux utilisateurs, qu'il s'agisse des clients ou des administrateurs, d'entrer les informations nécessaires dans le système pour une gestion efficace des tickets d'assistance.

### 7.2.3.1 Interfaces de saisie de données des clients

La figure ci-dessous illustre l'interface de saisie destinée aux Administrateurs dans le système de gestion de la plateforme d'assistance client. Cette interface permet aux administrateurs de soumettre les informations nécessaires pour créer un client.

The screenshot displays the 'NOVA LEAD' user management interface. A modal window titled 'Ajouter Un Utilisateur' is open, allowing an administrator to add a new user. The background shows a table of existing users and a list of actions (Details, Modifier, Archiver).

NOM	EMAIL
perez	mimi@gmail.com
COOPEC SIFA	kolaniflorent445@gmail.com
COOPEC AD	florentinoperez446@gmail.com

**Ajouter Un Utilisateur**

Nom \*  Email \*

Contact \*  Pays \*

Ville \*  Role \*

Figure 18 : Interface de saisie de données de client par l'administrateur

### 7.2.3.2 Interfaces de saisie de données des produits

La figure ci-dessous présente l'interface de saisie dédiée aux produits dans le système de gestion de la plateforme d'assistance Client. Cette interface permet de renseigner les informations nécessaires concernant les produits associés aux tickets d'assistance

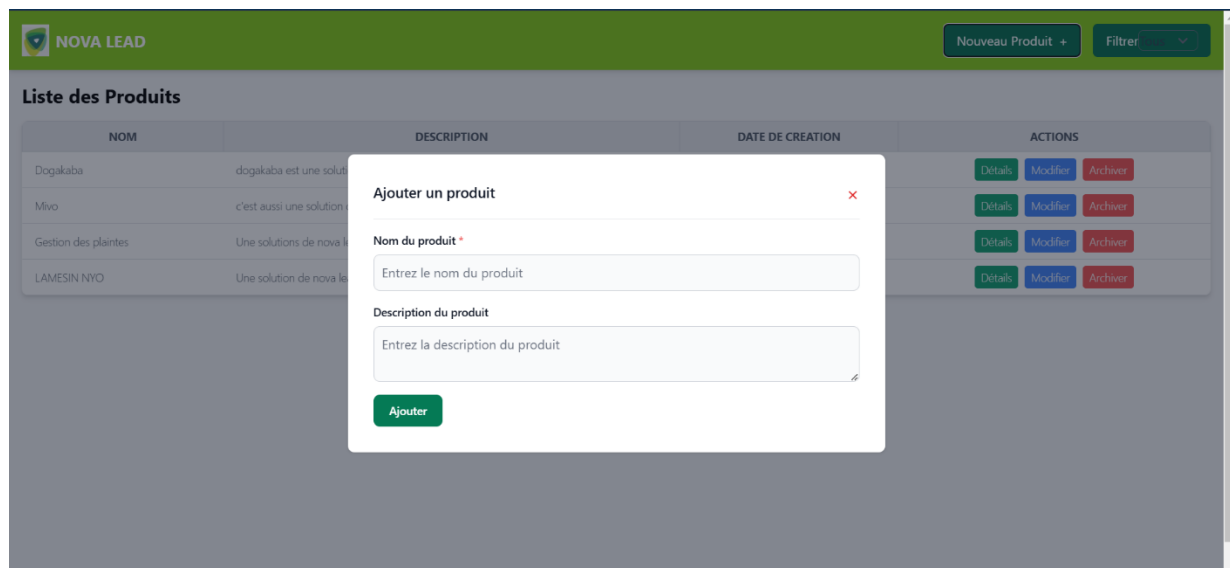


Figure 19 : Interface de saisie de données des produits

### 7.2.3.3 Interfaces de saisie de données des Types de demande

La figure ci-dessous illustre l'interface de saisie aux types de demande dans le système de gestion de la plateforme d'assistance Client. Cette interface permet de renseigner les informations nécessaires pour enregistrer un type de demande dans le système liées aux tickets d'assistance.

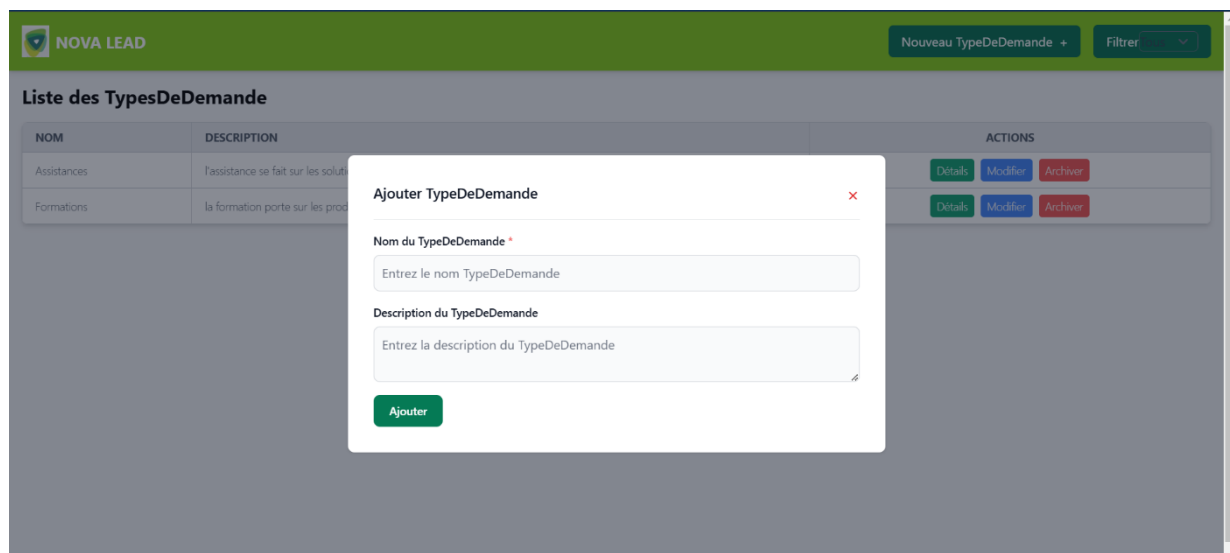


Figure 20 : Interface de saisie de données des types de demande

#### 7.2.3.4 Interfaces de saisie d'ajout de Tickets d'assistance

La figure ci-dessous présente l'interface de saisie dédiée à l'ajout de tickets d'assistance dans le système. Cette interface permet aux utilisateurs de soumettre une demande en fournissant les informations nécessaires, telles que le type de demande, le produit concerné, et les détails associés.

The screenshot displays the Nova Lead application interface. A modal window titled "Créer un Ticket" is open, allowing users to submit a new assistance request. The background shows a "Liste des Tickets" table with columns for Client, Type de Demande, Date de Création, Note, and Actions. The modal form includes the following fields:

- Produit \***: A dropdown menu with the placeholder "Sélectionner un produit".
- Type de Demande \***: A dropdown menu with the placeholder "Sélectionner un type de demande".
- Urgence \***: A dropdown menu with the placeholder "Sélectionnez l'Urgence".
- Description \***: A text input field.
- Télécharger un fichier (optionnel)**: A section with a "Choisir un fichier" button and the text "Aucun fichier choisi".
- Créer un ticket**: A green button at the bottom of the modal.

The background table "Liste des Tickets" shows three entries for "COOPEC AD" with types "Assistances", "Formations", and "Assistances". The "Date de Création" column shows "12/12/2024" for all entries. The "Note" column shows star ratings (4, 3, and 4 stars respectively). The "Actions" column shows a three-dot menu for each entry.

Figure 21 : Interface de saisie de données d'ajout de tickets



### 7.2.3.5 Interfaces d'échange entre le client et l'administrateur ou l'agent support

La figure ci-dessous illustre l'interface de conversation entre le client et l'administrateur ou l'agent support. Cette interface permet aux clients d'envoyer des informations concernant ses besoins ou demande.

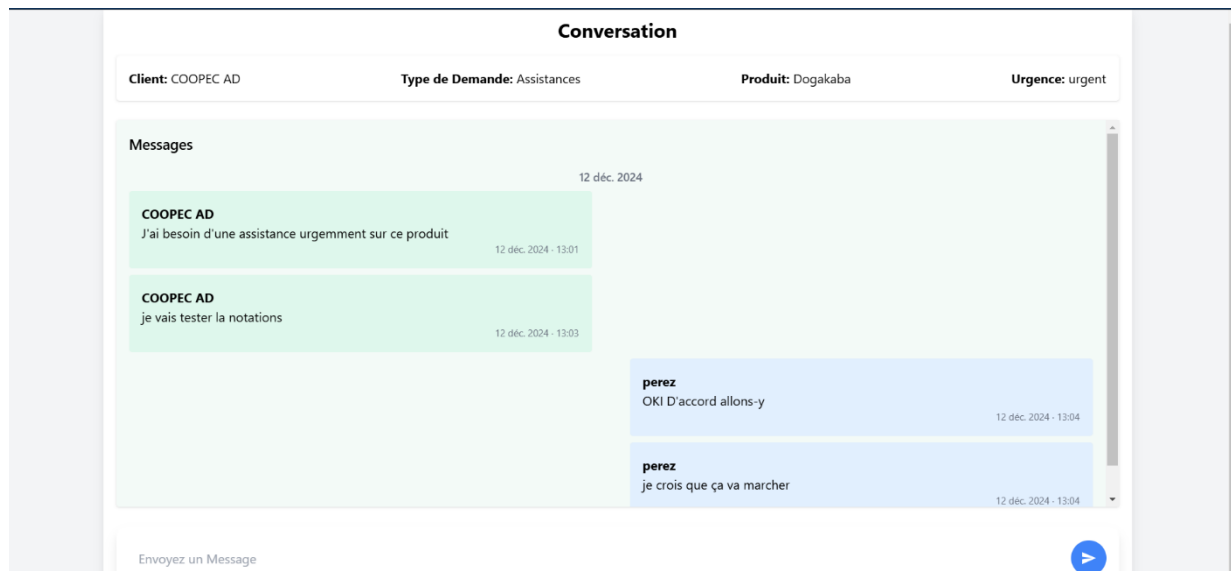


Figure 22 : interface d'échange entre le client et l'administrateur ou l'agent support

## CONCLUSION GÉNÉRALE

Ce document synthétise le travail effectué dans le cadre du développement de l'application d'assistance client pour NOVA LEAD, visant à répondre aux besoins d'une gestion efficace des interactions entre clients et administrateurs. L'analyse des méthodes actuelles, reposant souvent sur des échanges peu centralisés et une absence de suivi structuré, a mis en évidence des limites importantes comme la perte d'information, difficulté de gestion des demandes, et absence d'une vue d'ensemble sur les problèmes rencontrés par les clients.

Le projet a permis de répondre à ces problématiques en concevant une solution moderne basée sur Vue.js pour l'interface utilisateur, Node.js pour le backend, et MongoDB une base de données robuste pour centraliser les informations. Ce système facilite désormais la création, la gestion et le suivi des tickets, tant pour les administrateurs que pour les clients. Il offre des fonctionnalités adaptées telles que l'affichage des statuts des tickets, des détails sur les demandes, et un espace de réponse simplifié. L'intégration de l'authentification et des permissions garantit également un contrôle précis sur l'accès aux fonctionnalités.

Ce projet a été une opportunité enrichissante pour appliquer des connaissances techniques et développer des compétences pratiques, en répondant à des enjeux réels. Il a également permis d'explorer des outils modernes et de proposer une solution qui optimise les processus et renforce la qualité du service client.

En perspective, une évolution majeure serait l'ajout d'un module d'analyse et de reporting. Ce module permettrait de générer des statistiques sur les performances des réponses, le volume de tickets traités, et le temps de résolution moyen. Ces données aideraient à prendre des décisions éclairées pour améliorer les processus et offrir une expérience utilisateur encore meilleure. Avec cette évolution, l'application deviendrait une solution complète, couvrant à la fois la gestion des interactions clients et l'optimisation stratégique du service, tout en renforçant l'efficacité globale de la plateforme.

## Bibliographie

1. NOVA LEAD : Ingénierie Logicielle, développement de logiciel. *novalead.dev*. [En ligne] [Citation : 07 septembre 2024.] <https://www.novalead.dev>.
2. DOGA KABA. [En ligne] [Citation : 20 septembre 2024.] La plateforme digitale de gestion des demandes de crédit, des plaintes et réclamations, de la performance sociale et environnementale : Disponible :. <https://dogakaba.com/>.
3. LETAVI SMS (@letavitg)/X. *LETAVI*. [En ligne] [Citation : 30 septembre 2024.] <https://x.com/>.
4. Mivo CRM . [En ligne] [Citation : 26 septembre 2024.] une plateforme de décentralisation des activités opérationnelles de MIVO ENERGY, Disponible. MivoCRM.
5. Google Maps. *Google Maps*. [En ligne] [Citation : 02 septembre 2024.] [google.com/maps](https://google.com/maps).
6. Suite CRM Sellsy. *Suite CRM Sellsy*. [En ligne] [Citation : 25 Septembre 2024.] Logiciel CRM, facturation, pré-comptabilité Assistance Disponible:. [go.sellsy.com](https://go.sellsy.com).
7. Unified Modeling Language. *UML*. [En ligne] [Citation : 28 Septembre 2024.] Models and Diagrams for Beginners | Udemy Disponible : . [www.udemy.com](https://www.udemy.com).
8. *Cycle en V en gestion de projet : définition et méthode Disponible*. [En ligne] [Citation : 07 septembre 2024.] [manager-go.com](https://manager-go.com).
9. draw.io Disponible. *Draw.io*. [En ligne] [Citation : 17 Octobre 2024.] [drawio.com](https://drawio.com).
10. MongoDB. *MongoDB*. [En ligne] 8.0, 10 février 2009. [Citation : 05 Septembre 2024.] Documents consulter . <https://www.mongodb.com/fr-fr>.
11. Documentation. Disponible. *Git*. [En ligne] [Citation : 05 Octobre 2024.] <https://git-scm.com/doc>.

## TABLE DES MATIÈRES

DÉDICACE .....	1
REMERCIEMENT .....	2
RÉSUMÉ .....	3
ABSTRACT .....	4
SOMMAIRE .....	5
LISTE DES TABLEAUX .....	7
LISTE DES FIGURES.....	8
LISTE DES PARTICIPANTS .....	9
GLOSSAIRE.....	10
INTRODUCTION GÉNÉRALE .....	11
CHAPITRE 1 : PRÉSENTATION DU CADRE DE STAGE.....	12
1.1 Présentation de l'entreprise d'accueil : Nova Lead.....	12
1.2.1 Mission.....	12
1.2.2 Activités .....	12
1.2.3 Quelques réalisations .....	13
1.2.4 Plan de localisation .....	13
CHAPITRE 2 : CAHIER DES CHARGES.....	14
2.1 Contexte et justification du projet .....	14
2.2 Problématique .....	14
2.3 Objectifs .....	15
2.3.1 Objectif général.....	15
2.3.2 Objectif spécifique .....	15
2.4 Résultats attendus.....	15

2.5 Étude et critique de l'existant.....	16
2.5.1 Étude de l'existant.....	16
2.5.2 Critique de l'existant.....	16
2.6 Spécifications fonctionnelles .....	16
2.6.1 Gestion des clients .....	16
2.6.2 Gestion de demandes d'assistance .....	17
2.6.3 Suivi des tickets et communication.....	17
2.6.4 Gestion des utilisateurs .....	17
2.6.5 Suivi et rapports.....	17
2.7 Spécifications non fonctionnelles .....	17
2.8 Proposition et choix de solution.....	18
2.8.1 Première solution : Adoption d'un CRM .....	18
2.8.1.1 Sellsy.....	18
2.8.1.2 Odoo.....	19
2.8.2 Deuxième solution : Développement d'une application sur mesure.....	19
2.8.3 Choix de Solution.....	19
2.9 Planning de réalisation du projet.....	21
CHAPITRE 3 : ANALYSE ET CONCEPTION .....	22
4.1 Présentation de la méthode d'analyse et de conception .....	22
4.1.1 Méthode d'analyse .....	22
4.1.1.1 Présentation du langage de modélisation UML .....	23

4.1.2 Méthode de conception .....	23
4.1.2.1 Spécification des besoins utilisateurs.....	24
4.1.2.2 Test de recette .....	24
4.1.2.3 Spécification des exigences système .....	24
4.1.2.4 Test système .....	25
4.1.2.5 Conception globale (système).....	25
4.1.2.6 Test d'intégration.....	25
4.1.2.7 Conception détaillée.....	25
4.1.2.8 Test unitaire.....	25
4.1.2.9 Réalisation (implémentation).....	26
4.1.2.10 Tests finaux .....	26
4.2 Présentation de l'outils de modélisation .....	26
4.3 Etude détaillé de la solution.....	27
4.3.1 Identification des Acteurs .....	27
4.3.2 Diagramme de contexte statique .....	28
4.3.3 Diagramme des cas d'utilisation .....	28
4.3.3.1 Identification des cas d'utilisation par acteur .....	28
4.3.3.2 Schématisation .....	30
4.3.4 Description textuelle des cas d'utilisation .....	33
4.3.4.1 Enregistrer un client .....	33
4.3.4.2 Enregistrer un Produit .....	34

4.3.4.3 Enregistrer un type de demande.....	35
4.3.4.4 Enregistrer un Ticket.....	36
4.3.4.5 Enregistrer une Note .....	37
4.3.5 Diagramme d'activités.....	37
4.3.5.1 Enregistrer un client .....	38
4.3.5.2 Enregistrer un ticket .....	39
4.3.5.3 Ajouter une note .....	40
4.3.6 Diagramme des classes .....	41
CHAPITRE 4 : RÉALISATION ET MISE EN ŒUVRE.....	42
7.1 Mise en œuvre .....	42
7.1.1 Choix de matériel.....	42
7.1.1.1 Ordinateurs.....	42
7.1.2 Choix de logiciel.....	43
7.1.2.1 Environnement de développement.....	43
7.1.2.2 Système d'exploitation.....	43
7.1.2.3 Conception de l'interface utilisateur .....	43
7.1.2.4 Technologies de programmation.....	44
7.1.2.5 Langage de requête .....	45
7.1.2.6 Base de données.....	45
7.1.2.7 Architecture logicielle de l'application .....	45
7.1.2.8 Design Pattern de Node.js.....	47

7.1.2.9 Système de gestion et d'hébergement du code .....	47
7.1.3 Sécurité de l'application .....	47
7.1.3.1 Authentification des utilisateurs .....	48
7.1.3.2 Sécurité des données .....	48
7.1.4 Prévisions financières .....	48
7.2 Présentation de l'application .....	49
7.2.1 Présentation.....	49
7.2.1.1 Structure de l'espace administrateur .....	50
7.2.1.2 Structure de l'espace d'Agent Support .....	51
7.2.1.3 Structure de l'espace Client .....	52
7.2.2 Architecture de l'application.....	53
7.2.3 Quelques masques de saisies de la plateforme d'assistance client de Nova Lead..	54
7.2.3.1 Interfaces de saisie de données des clients.....	54
7.2.3.2 Interfaces de saisie de données des produits.....	54
7.2.3.3 Interfaces de saisie de données des Types de demande .....	55
7.2.3.4 Interfaces de saisie d'ajout de Tickets d'assistance .....	56
7.2.3.5 Interfaces d'échange entre le client et l'administrateur ou l'agent support .....	57
CONCLUSION GÉNÉRALE.....	58
Bibliographie.....	59
TABLE DES MATIÈRES .....	I



