



Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique
Direction Générale des Etudes Technologiques



Institut Supérieur des Etudes Technologiques de Djerba
Département TECHNOLOGIES de l'Informatique

Code projet	DSI04
----------------	-------

Rapport de
STAGE DE FIN D'ETUDES

Présenté en vue de l'obtention du diplôme de

**Licence Appliquée en TECHNOLOGIES de
l'Informatique**

Parcours : DSI

**DÉVELOPPEMENT D'UN SYSTÈME DE
GESTION POUR AGENCES IMMOBILIÈRES À
BASE DE MICROSERVICES**

Elaboré par :
Mohamed Kaba

Encadré par :
Mr Anis Asses

Effectué à :
Entreprise : McZEN TECHNOLOGIES
Encadrant : Mr Hamed Haouari

Année Universitaire : 2020/2021

DÉDICACES

Je dédie ce travail :

A mes chers parents

Zui ont œuvré pour ma réussite, de par leur amour inconditionnel, leur soutien, leurs prières et bénédictions, tous les sacrifices consentis pour mon instruction et mon bien-être et leur présence dans ma vie, que vous puissiez recevoir à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.

Puisse Dieu le très haut, vous accorder santé, bonheur et longue vie.

A toute ma famille

Aucun langage ne saurait exprimer mon respect et ma considération pour votre soutien et encouragements. Je vous dédie ce travail en reconnaissance de l'amour que vous m'offrez et de votre bonté exceptionnelle.

Que Dieu le Tout Puissant vous garde et vous procure santé et bonheur.

A Tata, Karamo, Alpha Mahmoud et mes amis.

 Mohamed KABA

REMERCIEMENTS

Je tiens avant toute chose à remercier Dieu le tout puissant qui m'a donné la force et la patience d'accomplir ce modeste travail.

En particulier, j'exprime ma profonde gratitude à Hamed Haouari pour m'avoir permis de bénéficier de ses conseils judicieux, de ses connaissances et de la confiance qu'il m'a témoignée qui ont fait de ce stage un moment très plaisant et intéressant au sein de McZEN TECHNOLOGIES.

J'adresse mes vifs remerciements à Mr Anis Asses pour son encadrement et ses conseils qui ont été d'un appui inestimable dans l'élaboration de ce travail. Je remercie aussi le personnel du département pour l'excellent accueil, l'enseignement, les conseils avisés et la bonne humeur partagée avec eux.

Également, je remercie la communauté guinéenne pour la joie, la drôlerie et les moments théâtraux et les repas partagés en famille.

Un grand merci à Aly Badara Condé, qui a été victime de mes multiples interrogations permanentes, et qui m'a également fait partager tout son savoir et son avoir.

Enfin, à toute personne ayant contribué de loin ou de près, je dis merci.

RÉSUMÉ

Ce présent rapport est le fruit de mon stage de fin d'étude. Il porte sur le projet intitulé « Développement d'un système de gestion pour agences immobilière à base de microservices »appelé BnB PRO visant à réaliser une application web.

Le développement de l'application a été fait suivant l'approche agile Kanban tout en s'inspirant de l'architecture JAVA EE (JAVA Entreprise Edition) qui tire profit de la souplesse des Frameworks Spring MVC, Spring Boot, Spring Security, Hibernate lors de la réalisation de la partie backend. Et aussi le Framework Vue.js pour les interfaces utilisateurs dans la partie frontend et enfin l'utilisation de MYSQL comme Système de Gestion de Bases de données (SGBD).

Abstract

This present report is the result of my end-of-study internship. It concerns the project entitled « Development of a management system for real estate agencies based on microservices »called BnB PRO aimed at creating a web application.

The development of the application was done following the agile Kanban approach while being inspired by the JAVA EE (JAVA Enterprise Edition) architecture which holds the advantage of the flexibility of the Spring MVC, Spring Boot, Spring Security, Hibernate Frameworks during the realization of the backend part. And the Vue.js Framework for the frontend user interfaces, finally the use of MYSQL like a Database Management System (DBMS).

Sommaire

INTRODUCTION GÉNÉRALE	1
1 ÉTUDE PREALABLE	3
1.1 Introduction	3
1.2 Mise en contexte	3
1.3 Présentation de l'organisme d'accueil	3
1.3.1 Présentation générale	3
1.3.2 Services	4
1.3.3 Valeurs offertes	4
1.4 Etude de l'existant	5
1.4.1 BnB PRO	5
1.4.2 Audit du site	5
1.4.3 Critiques de l'existant	7
1.4.3.1 Compte utilisateur	7
1.4.3.2 Evolutivité du système	7
1.4.3.3 Etat de la propriété	7
1.4.3.4 Expérience utilisateur	7
1.4.3.5 Localisation sur une carte	8
1.4.3.6 Code non optimisé	8
1.4.3.7 Création de contrat	9
1.4.3.8 Profil utilisateur	9
1.4.3.9 Gestion de la relation client (CRM)	9
1.4.4 Solution proposée	10
1.4.5 Conclusion	10
2 ÉTUDE DU PROJET ET ANALYSE DES BESOINS	11
2.1 Introduction	11
2.2 Présentation du projet	11
2.2.1 Fonctionnalités de l'application	11
2.2.2 Spécification des besoins	12
2.2.2.1 Besoins fonctionnels	12

3.4.2.1	Diagramme d'activité du cas d'utilisation « Inscriptio	31
3.4.2.2	Diagramme d'activité du cas d'utilisation « Ajouter une dépense »relatif à la tâche 5 « Gestion des dépenses »	32
3.5	Modélisation statique	32
3.5.1	Diagramme de classes	32
3.5.2	Diagramme de déploiement	33
3.6	Conclusion	34
4	DÉVELOPPEMENT DE L'APPLICATION	35
4.1	Introduction	35
4.2	Choix technologiques	35
4.2.1	Notion de Microservices	36
4.3	Environnement de travail	36
4.3.1	Environnement matériel	36
4.3.2	Environnement logiciel	37
4.3.2.1	Environnements de développement	37
4.3.2.2	Logiciels divers	37
4.3.3	Frameworks de développement	38
4.3.3.1	Frameworks Backend	38
4.3.3.2	Frameworks Frontend	40
4.3.4	Outils divers	41
4.4	Les principales interfaces graphiques	42
4.4.1	Page d'authentification	42
4.4.2	Page de création de compte pour une agence	43
4.4.3	Gestion de la propriété	44
4.4.4	Création du modèle de contrat	46
4.5	Autres caractéristiques particulières de l'application	47
4.5.1	Persistance des données	47
4.5.2	Sécurité	48
4.5.3	Cartographie avec Leaflet	48
4.5.4	Charte graphique	48
4.5.5	Editeur WYSIWYG Ckeditor	49
4.6	Conclusion	49
CONCLUSION GÉNÉRALE ET PERSPECTIVES		50
BIBLIOGRAPHIE ET NÉTOGRAPHIE		52

Annexes	53
Annexe A : Les pratiques de Kanban	54
Annexe B : Création de projet et de fiches Kanban avec Jira	55
Annexe C : Référencement du site au niveau de différents moteurs de recherche.	60
Annexe D : Description du fonctionnement de la persistance de données au niveau de Spring.	62
Annexe E : Communication avec les services via Postman.	64

Table des figures

1.1	Tableau de bord de BnB PRO	5
1.2	Capture d'écran du résultat de Dareboost	6
1.3	Décalage entre les tailles et l'alignement des colonnes	8
1.4	En cliquant sur afficher les infomations d'un client, ça redirige vers l'ajout d'un nouveau client et l'affichage des informations et sur la même page.	8
1.5	Capture d'écran d'un modèle de contrat	9
2.1	Structure générale du projet	14
2.2	Structure de l'application	15
2.3	Tableau Kanban à la date du 25/05/2021	19
2.4	Diagramme de GANTT	21
3.1	Diagramme des cas d'utilisation général	23
3.2	Diagramme du cas d'utilisation « <i>gérer les clients</i> »	25
3.3	Fonctionnement de l'architecture MVC	28
3.4	Diagramme de séquence d'Authentification	29
3.5	Diagramme de séquence « <i>gérer les dépenses</i> »	30
3.6	Diagramme de d'activité « <i>inscription</i> »	31
3.7	Diagramme de d'activité « <i>ajouter une dépense</i> »	32
3.8	Diagramme de classes relatif à la tâche 6 « <i>Gestion des tâches</i> » . .	33
3.9	Diagramme de déploiement général	34
4.1	Page d'authentification	43
4.2	Page de création de compte pour une agence	44
4.3	Page de gestion de la propriété	45
4.4	Page de création de la propriété	45
4.5	Page de création du modèle de contrat	46
4.6	Description du fonctionnement de la persistance des données . . .	47
4.7	Description du mécanisme JWT	48
8	Sélection et description des type de projet Jira	55

Table des figures

9	Sélection et description des type de projet Jira	56
10	Choix de la méthodologie Agile	57
11	Nom et création du projet	58
12	Tableau de bord Jira	59
13	BnB PRO sur Google	60
14	BnB PRO sur Yahoo !	61
15	BnB PRO sur Bing	61
16	Classe "Client"	62
17	Interface "ClientRepository"	63
18	Fichier de configuration "application.properties"	63
19	La methode POST	64
20	La methode GET	65
21	La methode PUT	65
22	La methode DELETE	66

Liste des tableaux

1.1	Fiche technique de McZEN TECHNOLOGIES	4
2.1	Tableau comparatif entre les méthodes traditionnelles et agiles	17
2.2	Tableau comparatif entre Scrum et Kanban	18
2.3	Chronogramme des tâches	20
3.1	Description du cas d'utilisation « S'authentifier »	24
3.2	Description du cas d'utilisation « <i>ajouter un collaborateur</i> »	24
3.3	Description du cas d'utilisation « <i>Ajouter un client</i> »	25
3.4	Description du cas d'utilisation « <i>modifier un client</i> »	26
3.5	Description du cas d'utilisation « <i>supprimer un client</i> »	26
3.6	Description du cas d'utilisation « <i>rechercher un client et l'afficher</i> »	27
4.1	Descriptif de la notion de Microservices	36
4.2	Descriptif des éditeurs utilisés	37
4.3	Descriptif des logiciels utilisés	38
4.4	Tableau 1 : descriptif des Frameworks utilisés pour le Backend	39
4.5	Tableau 2 : descriptif des Frameworks utilisés pour le Backend	40
4.6	Descriptif des Frameworks utilisés pour le frontend	41
4.7	Descriptif de quelques outils utilisés	42
4.8	La charte graphique	49

INTRODUCTION GÉNÉRALE

Le développement du numérique entraîne de plus en plus de professionnels de l'immobilier à se tourner vers des solutions de gestion leur permettant de gérer leur activité. Pour une agence immobilière, un intermédiaire ou un particulier, une application de gestion immobilière est aujourd'hui le partenaire idéal étant professionnel. Les entreprises dans ce milieu l'ont bien compris. C'est pour cela que de nombreux logiciels sont aujourd'hui présents sur le marché.

Cependant, il peut être compliqué de dénicher l'application la mieux à ce secteur et aux activités.

Il convient d'abord de rappeler qu'une solution de gestion pour agences immobilières, c'est un logiciel ou une application web ou mobile qui permet, en tant que professionnel de l'immobilier, de mener à bien les principales activités, de la prospection des clients à la finalisation de la transaction. Elle est tout autant un outil de gestion que de communication et de développement des activités des agences. L'organisme au sein duquel j'ai réalisé ce projet de fin d'étude dénommé McZEN TECHNOLOGIES dispose d'une plateforme de gestion[1] destinée à ces agences et professionnels de l'immobilier mettant à leur disposition une application web pour leur permettre de gérer et suivre leur activité où qu'ils soient et en temps réel.

Elle fait partie d'une solution complète qu'offre cette société qui comprend aussi un site web immobilier[2] et une multidiffusion des annonces.

Le problème c'est que réellement, les agents immobiliers veulent un système qui les aide à vendre alors que le système actuel les aide à s'organiser. Ce qui fait conclure qu'ils payent les services de celui-ci plus pour le site web immobilier qui y est lié (mentionné ci-dessus, il permet de faire les annonces immobilières associé au système) que pour la plateforme de gestion en elle-même.

C'est dans cette optique d'idée que s'inscrit ce projet qui consiste à développer un système innovant, ergonomique, puissant et surtout beaucoup plus complet de gestion pour les agences immobilières.

L'objectif au niveau de ce projet de fin d'étude reste de faire une application web à base de Microservices qui permettra de centraliser la gestion totale des agences,

INTRODUCTION GÉNÉRALE

d'automatiser les tâches répétitives et apporter une certaine souplesse au sein de l'agence car elle se concentrera davantage sur son métier c'est-à-dire consacrer plus de temps aux clients, gérer les biens immobiliers en leur possession...

Ce rapport présente donc l'ensemble des étapes suivies pour la réalisation de cette solution. Il contient quatre chapitres organisés comme suit :

Le premier chapitre intitulé « Etude préalable » est consacré à la présentation du cadre de travail, du projet ainsi que de l'organisme d'accueil et l'étude de l'existant.

Le chapitre « Étude du projet et analyse des besoins » s'articule entre autre autour de la présentation du projet, l'identification des besoins fonctionnels et non fonctionnels de cette application et la planification.

Le troisième chapitre intitulé « Étude conceptuelle » détaille les différents aspects conceptuels de l'application.

Enfin le dernier chapitre intitulé « Développement de l'application » présente l'environnement de travail ainsi que les outils logiciels qu'on a utilisé pour la réalisation du projet. Il illustre aussi le travail réalisé avec un ensemble d'interfaces graphiques conçues pour l'application.

En conclusion, on mentionnera les différents atouts de ce projet et les perspectives d'améliorations possibles.

Chapitre 1

ÉTUDE PREALABLE

1.1 Introduction

L'étude préalable constitue une étape préliminaire pour la réalisation d'une application.

En effet, elle permet d'analyser, d'évaluer et de critiquer le fonctionnement habituel, tout en indiquant une solution

Dans ce chapitre, on va commencer par présenter l'entreprise d'accueil et ses services, ensuite le cadre du travail pour passer ensuite à l'étude de l'existant, les problématiques et enfin la solution proposée.

1.2 Mise en contexte

Ce projet s'inscrit dans le cadre d'un stage de fin d'études pour l'obtention d'un diplôme de Licence Appliquée en TECHNOLOGIES de l'Informatique parcours Développement des Systèmes d'Information de l'Institut Supérieur des Etudes Technologiques (ISET) de Djerba.

Ce stage est effectué au sein d'une société d'aide à la transformation digitale des entreprises à savoir McZEN TECHNOLOGIES.

1.3 Présentation de l'organisme d'accueil

1.3.1 Présentation générale

McZEN TECHNOLOGIES[3] est une société qui offre des services en tant que CTO[4] (Chief Technical Officer - ou Directeur des Nouvelles TECHNOLOGIES). Fondée en 2019, elle a son bureau à la pépinière d'entreprise de Djerba.

Grâce à l'effort de son équipe formée de jeunes doués, dévoués et passionnés de

nouvelles TECHNOLOGIES, la société ne cesse pas de s'agrandir.

1.3.2 Services

McZEN TECHNOLOGIES offre un savoir faire reconnu dans la mise en œuvre de projets couvrant :

- Planification et validation de la pile technologique.
- Ingénierie des produits et mise en place des processus qualité, sélection des outils.
- Validation et obtention de partenaires d'externalisation à distance.
- Préparation et validation de l'embauche interne.
- Le service se terminera par l'embauche d'un directeur technique interne.
- Validation et présentation aux investisseurs providentiels.

1.3.3 Valeurs offertes

- Minimiser les risques techniques liés à la croissance future et à la planification de la gamme de produits.
- Apprendre et faciliter le transfert de technologie des professionnels de votre domaine.
- Passez rapidement à l'échelle supérieure et ne pas s'enliser pas dans un processus d'embauche long et coûteux.
- Obtenez des conseils et de l'aide pour recueillir des fonds pour votre démarrage.

Dénomination Social	McZEN TECHNOLOGIES
Date d'implantation	Juin 2019
Siège Social	Pépinière d'entreprise de Djerba
Forme Juridique	Société À Responsabilité Limitée (SARL)
Secteur D'activité	Informatique : gestion des produits et transformation digitale des entreprises
Effectif Total	5 personnes
Contact	-Téléphone : +21620179154 -Site : https://mc-zen.com/ -Email : contact@mczen.com/

TABLE 1.1 – Fiche technique de McZEN TECHNOLOGIES

1.4 Etude de l'existant

On ne saurait débuter ce travail sans avoir une idée claire et précise sur l'existant quel qu'il soit.

La première tâche a donc été de rencontrer le responsable chargé de la gestion de la société. C'est à l'issue de cette rencontre qu'on a pu avoir une connaissance large de la plateforme créée par l'entreprise appelée BnB PRO et qu'on décrira dans les lignes qui suivent.

1.4.1 BnB PRO

Cette application web réalisée par ladite société s'adresse aux agences immobilières ou aux professionnels de l'immobilier en mettant à leur disposition une application web 100% cloud qui leur permet de gérer les propriétés, contrats, paiements reçus, dépenses et surtout suivre leur activité et rester en contact avec les clients à tout moment et n'importe où. Comme déjà mentionné, elle fait partie intégrante d'une solution complète de gestion pour agences immobilières qui se compose aussi un site web immobilier personnalisable et une multidiffusion (partage des annonces sur votre site, vos groupes et votre page Facebook, etc...).

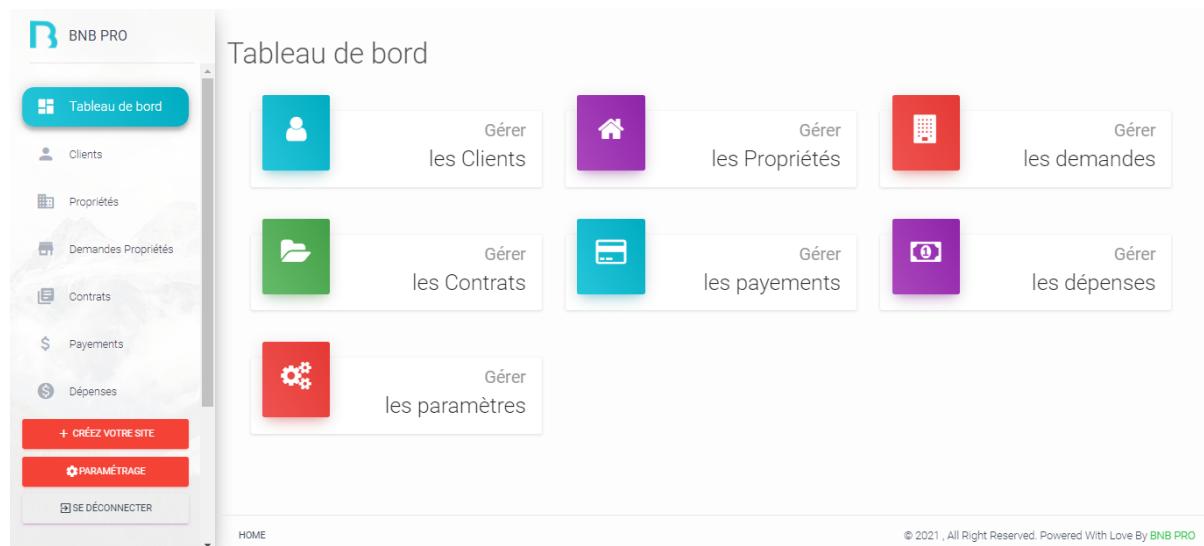


FIGURE 1.1 – Tableau de bord de BnB PRO

1.4.2 Audit du site

Lors de l'audit du site en question, j'ai utilisé plusieurs services permettant d'analyser les qualité et performance des sites webs, en particulier Dareboost[5] dont voici le résultat :

- La note : 68%
- Speed Index¹ : 8,309s
- Chargement terminé après : 3,889s
- Le poids : 1,85 Mo
- JavaScript : 94% du poids total
- CSS : 3% du poids total
- Autres : 3% du poids total

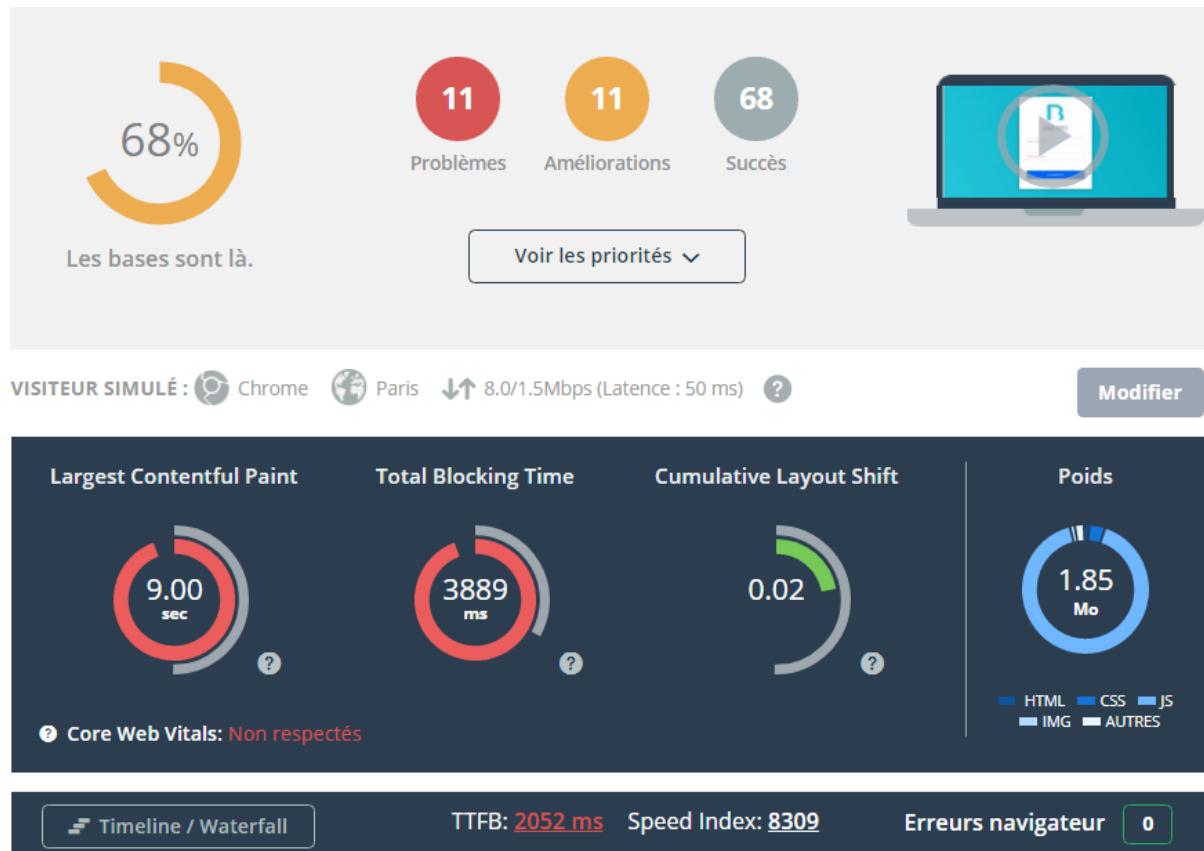


FIGURE 1.2 – Capture d'écran du résultat de Dareboost

Les points positifs qu'on peut lister sont les suivants :

- ✓ Le site est bien référencé sur les moteurs de recherches (Voir Annexe C 60).
- ✓ Le design simpliste avec une bonne lisibilité du contenu qui est d'ailleurs facile à trouver.
- ✓ L'ergonomie du site est pas mal, l'affichage des interfaces et les images s'adaptent quelque soit l'appareil (responsivité) et la vitesse de chargement des pages est bonne (< 15 s).

1. C'est un indice de performance de chargement de page qui indique à quelle vitesse le contenu d'une page commence à être visible. Plus l'affichage de cette zone est rapide, plus le Speed Index est petit.

- ✓ La sécurité du site est bien renforcée car il est équipé d'un certificat SSL et toutes les ressources sont chargées sur HTTPS.

1.4.3 Critiques de l'existant

1.4.3.1 Compte utilisateur

Le système actuel ne permet pas la création de plusieurs comptes pour une agence. Il ne met à la disposition qu'un seul compte utilisateur par agence et pour toutes les opérations là-dessus.

1.4.3.2 Evolutivité du système

Cette plateforme, réalisée avec Spring Boot dans sa version 1.9, Yii 2.0.15 et Angular 4, se confronte à un problème d'évolutivité car ces versions sont dépassées (Spring Boot 2.4.3, Yii 2.0.4 et Angular 11.2.5 à l'heure où j'écris ces textes). D'autant plus que la société ne maintient plus ses connaissances en Angular.

1.4.3.3 Etat de la propriété

Au niveau de la gestion des propriétés, dès qu'une nouvelle propriété est ajoutée, elle sera automatiquement partagée et publiée. La plateforme ne permet la définition de l'état d'une propriété (visibilité active ou non), ce qui peut être un inconvénient.

1.4.3.4 Expérience utilisateur

Les interfaces homme-machine (IHM) ne permettent pas une bonne expérience d'utilisation. Elles sont certes simplistes mais pas attractives.

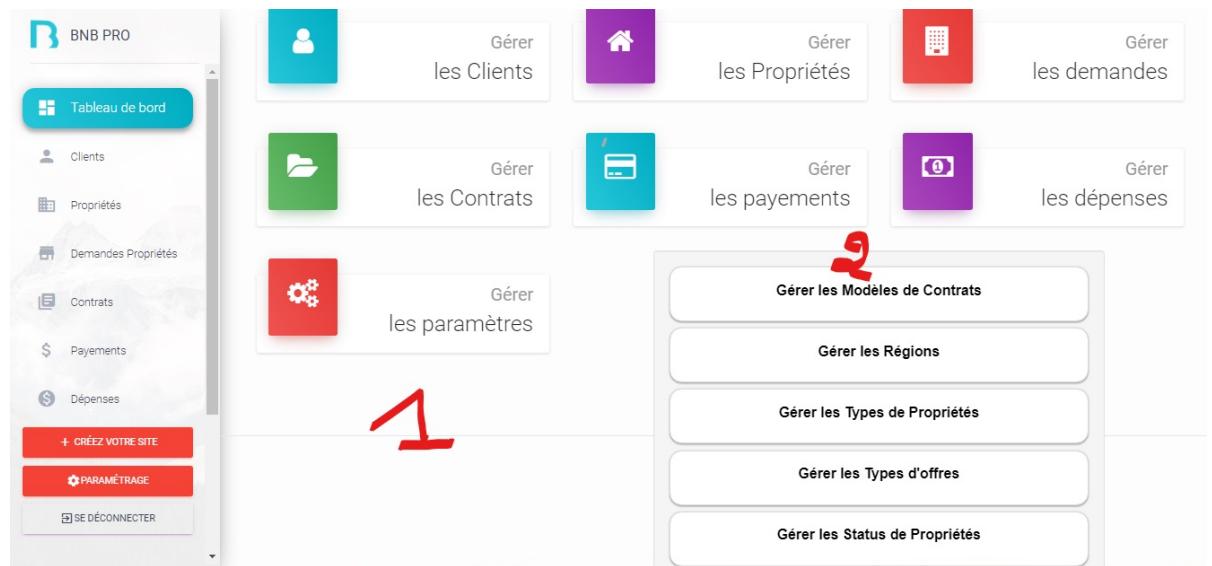


FIGURE 1.3 – Décalage entre les tailles et l’alignement des colonnes

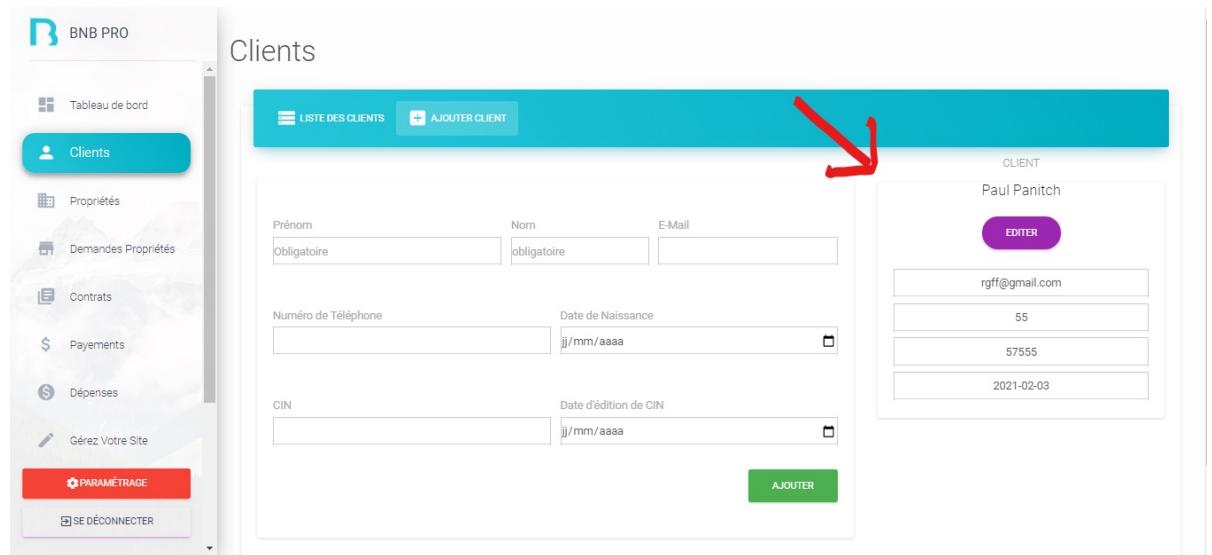


FIGURE 1.4 – En cliquant sur afficher les infomations d’un client, ça redirige vers l’ajout d’un nouveau client et l’affichage des informations et sur la même page.

1.4.3.5 Localisation sur une carte

On ne peut pas localiser la propriété sur la carte car cette fonctionnalité est inexistante.

1.4.3.6 Code non optimisé

Étant donné qu'il existe des utilisateurs qui ont des centaines propriétés par exemple, à chaque requête les concernant il les appelle toutes et ça ralentit le système. Avec le temps, les utilisateurs remarquent que leurs interfaces sont de

plus en plus lourdes. C'est dû au fait que le code n'est pas optimisé.

1.4.3.7 Crédation de contrat

En principe la création de modèle de contrat permet de définir un modèle personnalisé associé à une transaction (vente ou location) réalisée par l'agence directement sur le site sans passer par d'autres logiciels exemple Word. En pratique cette notion est très compliquée car l'éditeur de HTML de type WYSIWYG² n'est ni intuitif et ni attractif.

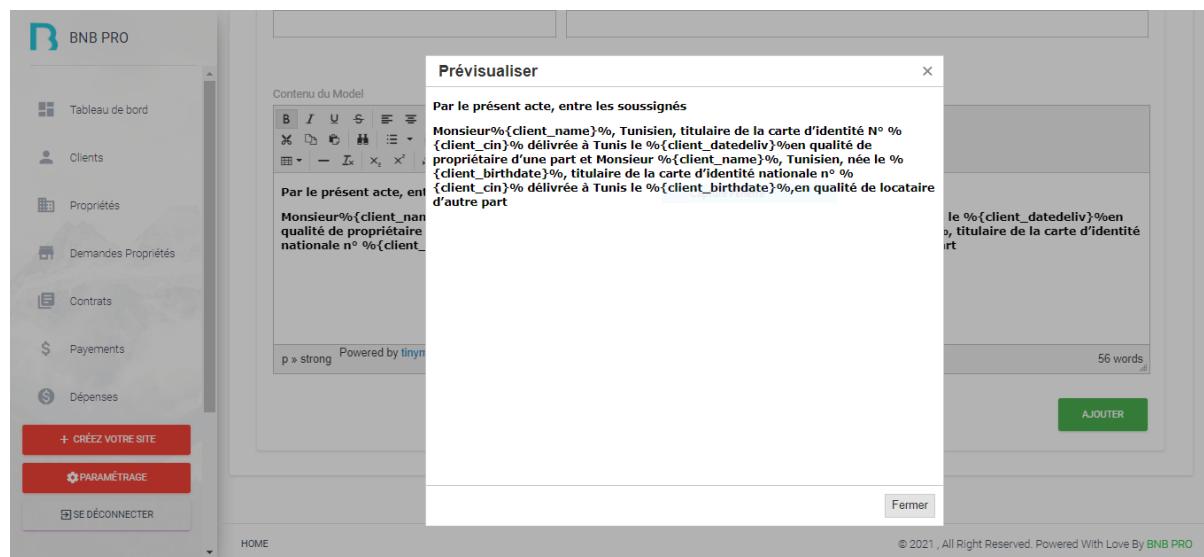


FIGURE 1.5 – Capture d'écran d'un modèle de contrat

1.4.3.8 Profil utilisateur

Ce système ne permet pas la définition du mot de passe par l'utilisateur lors de la création du compte, il lui octroie un mot de passe par défaut. Et aussi ne permet pas la modification du profil utilisateur ce qui est quand même une fonctionnalité basique.

1.4.3.9 Gestion de la relation client (CRM)

Le système a essayé de toucher tout ce qui est gestion interne d'une agence mais n'a pas touché ce que les agences veulent encore de plus qui est la gestion de la relation client à travers :

La gestion des tâches : Qui est une des composantes essentielles d'un CRM immobilier car elle permet d'organiser le travail, de donner la possibilité

2. « What You See Is What You Get » est une interface graphique qui permet à l'utilisateur de voir son document tel qu'il sera publié

de suivre l'historique des actions associé à un client, de programmer des notifications (rappel de paiement, contact, etc) et plein d'autres choses.

Phase de cycle de vie : Une autre composante très importante est la suivie du cycle de vie des clients : d'un prospect à un client véritable.

La gestion des campagnes marketings : Un des but de toutes agences est d'augmenter sa notoriété, améliorer sa visibilité et faire promouvoir ses biens auprès des clients potentiels pour atteindre leur objectifs business.

1.4.4 Solution proposée

En tenant compte des critiques de l'existant, on est amené à proposer une solution qui répond aux objectifs et pallier aux lacunes constatées au niveau du processus existant.

On propose alors de concevoir et développer une application web de gestion pour les agences immobilières à base de Microservices qui assure la disponibilité des informations en tout temps et avec n'importe quel moyen (responsive³) mais aussi une meilleure sécurité en termes d'accès sans oublier une interface ergonomique offrant une bonne expérience utilisateur.

L'application est composée d'une partie frontend⁴ réalisé avec le Framework⁵ Vue.js et une partie backend⁶ réalisée en utilisant entre autres le Framework Spring Boot qui interagit avec une base de données MySQL[7].

1.4.5 Conclusion

Dans ce chapitre, on a présenté le contexte général du projet, en introduisant l'organisme d'accueil. On a ensuite effectué une étude de l'existant, grâce à laquelle on a pu cerner les problématiques qui se présentent et enfin proposée une solution.

3. Permet de modifier la mise en page du site afin que le contenu s'adapte à l'écran quel que soit le terminal utilisé (smartphone, tablette, ordinateur de bureau, etc...)

4. Le terme « frontend » désigne les éléments d'un site que l'on voit à l'écran et avec lesquels on peut interagir depuis un navigateur.

5. Framework[6] signifie "cadre de travail", il permet aux développeurs d'être plus efficaces dans la conception d'applications web, entre autres, en offrant une architecture et des composants logiciels prêts à l'emploi et réutilisables.

6. Le « backend » est un peu comme la partie immergée d'un iceberg. On ne la voit pas en tant que simple internaute mais elle représente une très grande partie d'un projet web.

Chapitre 2

ÉTUDE DU PROJET ET ANALYSE DES BESOINS

2.1 Introduction

On procéde dans ce chapitre à la présentation du projet, ses objectifs, ensuite à la spécifications des besoins du cahiers de charges pour mieux comprendre le travail demandé en dégageant les besoins des différents utilisateurs que le système doit accomplir et enfin détailler la planification de celui-ci et sa méthode de développement.

2.2 Présentation du projet

De nos jours, il existe sur le marché tunisien très peu d'applications destinées aux professionnels de l'immobilier et de plus, il est très difficile d'en trouver une qui combine toutes les fonctionnalités adéquates.

Ce projet a pour but de réaliser un système complet de gestion qui cible principalement les agences immobilières mais aussi les particuliers ou intermediaires de ce secteur. Ce sera une application web qui permettra de tout gérer de A à Z, c'est-à-dire des biens en vente ou en location aux contrat en passant par la relation client et autres. Cette solution a pour rôle d'aider à la gestion des affaires afin de faciliter le travail au quotidien et ainsi offrir un gain de temps et d'efficacité considérable.

2.2.1 Fonctionnalités de l'application

L'application web à réaliser comporte plusieurs fonctionnalités dont :

- La gestion de clients qui permet de gérer et savoir les clients ou les partenaires avec lesquels l'agence travaille.

- La gestion de propriétés dont l'insertion de nouveaux biens immobiliers, leur modifications et aussi la consultation de ceux sur lesquels l'agence travaille.
- La gestion de demandes de propriétés qui permet de gérer les demandes des clients en vue de les satisfaire.
- La gestion de contrats pour la création de modèle de contrat personnalisé associé à une transaction réalisée par l'agence.
- La gestion de paiements qui permet d'historiser les différentes facturations avec leurs modes de paiement.
- La gestion de dépenses qui va permettre de connaître toutes les charges que l'agence a dépensées.
- La gestion de tâches qui permet à l'utilisateur d'avoir une vision globale sur les différentes tâches à effectués et à s'organiser.

2.2.2 Spécification des besoins

La spécification est la première étape dans un projet. Cette étape est déterminante pour le bon déroulement du projet. Elle consiste à connaître le travail demandé et les différents problèmes posés par le sujet du point de vue organisationnel et technique. On commencera par une présentation de la spécification générale de ce projet de point de vue besoins fonctionnels et non fonctionnels.

2.2.2.1 Besoins fonctionnels

Un besoin fonctionnel est un besoin spécifiant une action qu'un système doit être capable d'effectuer sans aucune contrainte physique : besoins spécifiant un comportement d'entrée/sortie d'un système.[8]

Cette application sera composé de plusieurs modules de gestion permettant des opérations basiques telles que le SCRUD (Search, Create, Read, Update et Delete). Ces modules sont entre autre la gestion :

- de clients
- de propriétés
- de demandes de propriétés
- de contrats
- de paiements
- de dépenses
- de tâches
- de paramétrages

2.2.2.2 Besoins non fonctionnels

Après avoir déterminé les besoins fonctionnels on présente ci-dessous l'ensemble des contraintes à respecter pour garantir la performance du système, donc de fournir un produit performant qui respecte les exigences de l'utilisateur et qui peut faire face à des risques de panne ou de non fonctionnement.

La fiabilité : l'application doit répondre aux besoins fonctionnels cités correctement lors de son interrogation.

La disponibilité : L'application doit être disponible à tout instant et être utilisée par n'importe quel utilisateur.

La modularité de l'application : Avoir un code simple facile à maintenir et à comprendre en cas de besoin.

Extensibilité : Afin de lui permettre de survivre assez longtemps, le caractère extensible de cette application doit être pris en considération.

L'ergonomie de l'interface : L'interface de l'application doit être conviviale et claire pour les utilisateurs quel que soit le moyen d'accès.

Sécurité : L'application porte des informations personnelles alors il faut s'assurer de la sécurité de ses informations.

2.2.3 Identification des acteurs

Cette section a pour objectif de présenter les acteurs et les fonctionnalités auxquelles doit répondre cette application.

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système.

Administrateur :

C'est l'un des acteurs principaux qui interagissent avec cette application. Il appartient à une agence immobilière ou c'est un professionnel de l'immobilier et il pourra faire la gestion :

- de clients
- de propriétés
- de demandes de propriétés
- de contrats
- de paiements
- de dépenses
- de tâches

- de paramétrages
- En outre, il pourra
- Éditer son profil.
- Ajouter un nouveau collaborateur.

Le collaborateur

C'est un employé d'une agence, il est ajouté par l'administrateur et fait la même chose que lui sauf ajouter un nouveau collaborateur.

2.2.4 Architecture de l'application

D'une manière générale, le projet sera divisé en deux parties : la partie frontend et la partie backend.

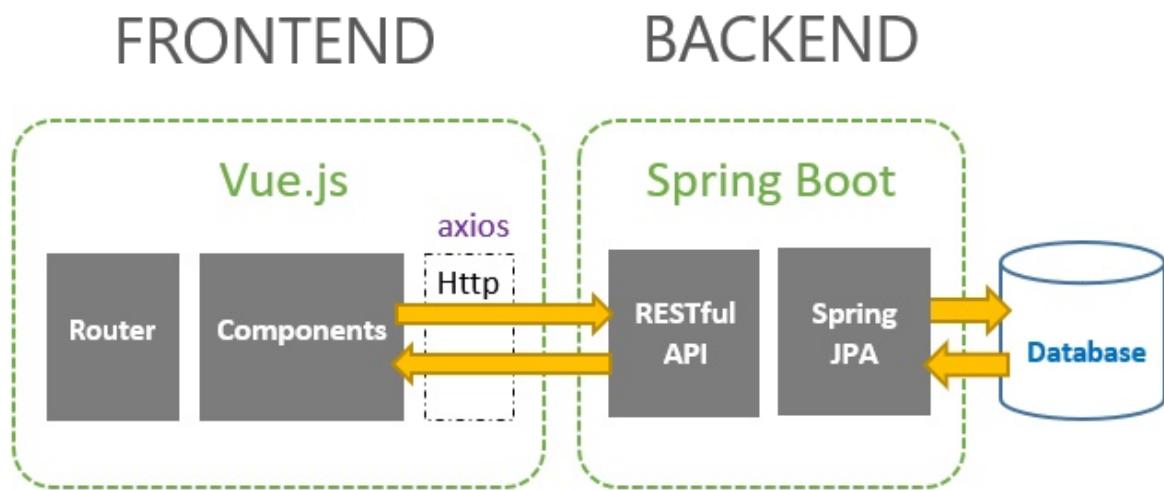


FIGURE 2.1 – Structure générale du projet

Schématiquement (voir la figure suivante), la structure de l'application se présente suit :

- Le client qui représente l'appareil utilisé (ordinateur, smartphone, etc..) va se connecter au site.
- La requête sera envoyée à la couche Contrôleur qui va la réceptionner, la rediriger vers la couche Service et fournir la réponse au client.
- Dans la couche Service seront définies les différentes méthodes et ce sont elles qui vont exécuter les traitements métiers.
- La couche Modèle contiendra les objets métiers.
- Et enfin la couche Repository qui communiquera avec la base de données.

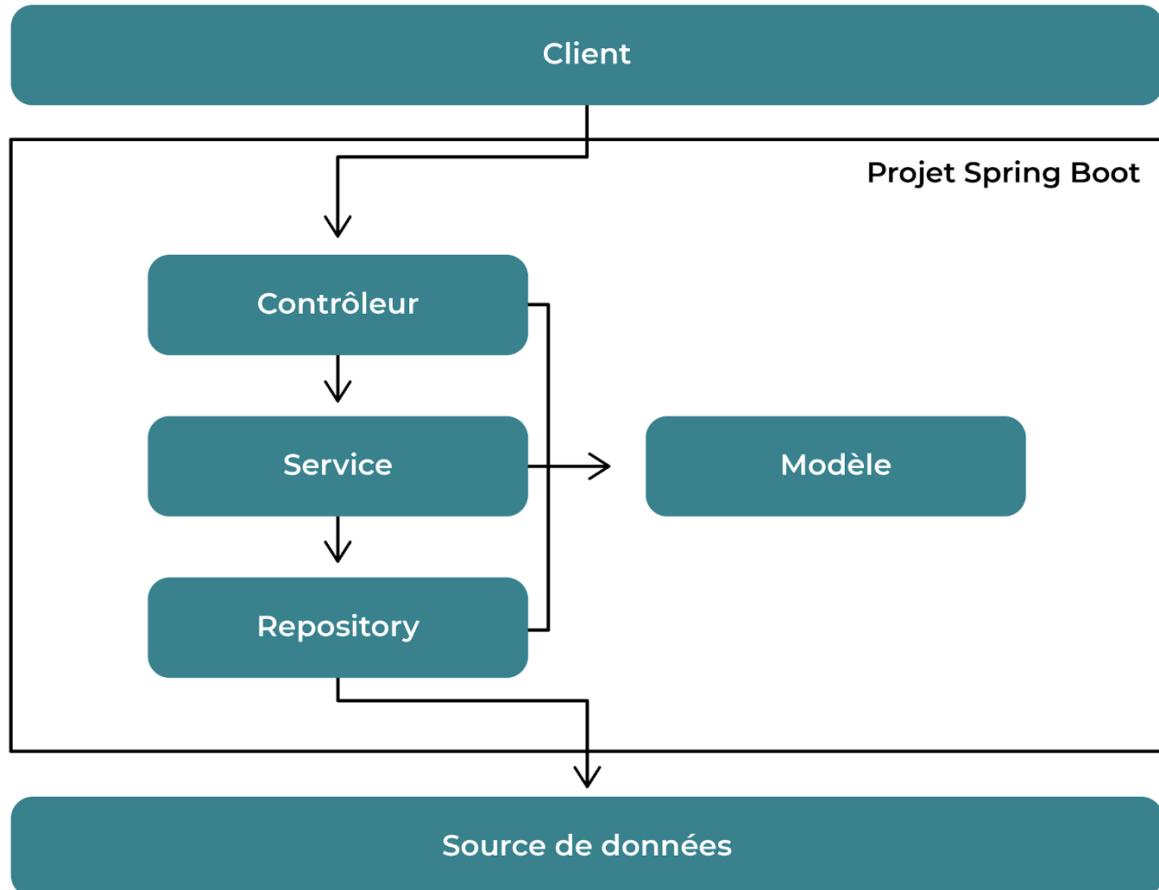


FIGURE 2.2 – Structure de l'application

2.2.5 Conduite de travail

Le projet comprend deux phases : la première est la recherche d'une solution convenable pour réaliser l'application et la deuxième est la conception et le développement.

2.2.5.1 Phase de recherche

C'est l'étape qui inclut l'étude bibliographique, dans laquelle on doit saisir les différentes notions et TECHNOLOGIES à utiliser dans le projet, les architectures, etc. Aussi, elle renferme les tests des différentes solutions mises en hypothèse pour réaliser l'application, et on fixe les outils nécessaires pour la réalisation du projet.

2.2.5.2 Phase de conception et développement

C'est une étape dans laquelle on modélise le système à réaliser en spécifiant les besoins fonctionnels pour clarifier les tâches à accomplir dans la partie développement. Cette phase se termine par une partie qui comprend la programmation et

les tests de validation.

2.2.6 Objectifs de l'application

L'application permettra de :

- Centraliser la gestion totale de leur agence immobilière.
- Être organisé et permettre la visualisation des différentes activités à effectués.
- Automatiser les tâches répétitives comme la création des contrats & les quittances de paiements, etc...
- Être toujours disponible, toujours à jour. Donnant la liberté totale d'utiliser l'application quel que soit l'appareil ou l'emplacement.
- Gérer les documents de paiements et de dépenses en liaison avec les clients et les biens immobiliers dans un seul espace sécurisé.
- Suivre le cycle de vie des clients et aider à vendre.

2.3 Planification du projet

La planification du projet est une étape importante d'avant-projet. Elle consiste à prévoir le déroulement du projet tout au long des phases constituant le cycle de développement.

2.3.1 Méthodologie et formalismes adoptés

Une méthode d'analyse et de conception est un procédé qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client. Pour ce faire, on part d'un énoncé informel (le besoin tel qu'il est exprimé par le client, complété par des recherches d'informations auprès des experts du domaine fonctionnel, comme les futurs utilisateurs d'un logiciel), ainsi que de l'analyse de l'existant éventuel (c'est-à-dire la manière dont les processus à traiter par le système se déroulent actuellement chez le client).[9]

Il existe deux grandes familles de méthodes : méthodes traditionnelles et méthodes agiles.

2.3.1.1 Comparaison entre les deux méthodes

Il est très important de bien choisir la méthodologie qui répond parfaitement à ses besoins, s'adapte au mieux aux exigences du client et qui fournit l'ouvrage dans les plus brefs délais. Pour cela, il faut effectuer une étude comparative entre

les deux méthodes afin d'en sélectionner la meilleure qui s'adapte au projet. Le tableau suivant décrit les caractéristiques de chaque méthode.

Méthodes traditionnelles	Méthodes agiles
En cascade ou en V phases séquentielles	Itératif et incrémental
Planification : Prédictive	Planification : adaptative
Documentation : Produite en quantité importante	Documentation : Réduite au strict nécessaire
Une équipe avec des ressources spécialisées, dirigées par un chef de projet	Une équipe responsabilisée où l'initiative et la communication sont privilégiées, soutenue par le chef de projet
Contrôle qualité à la fin du cycle de développement. Le client découvre le produit fini	Un contrôle qualité précoce et permanent, au niveau du produit et du processus. Le client visualise les résultats tôt et fréquemment
Résistance changement, Processus lourds de gestion de changements acceptés	Accueil favorable au changement inéluctable, intégré dans le processus
Suivi de l'avancement : Mesure de la conformité aux plans initiaux. Analyse des écarts	Un seul indicateur d'avancement : le nombre de fonctionnalités implémentées et le travail restant à faire
Processus distinct, rigoureux, de gestion des risques	Gestion des risques intégrée dans le processus global
Mesure de succès : Respect des engagements initiaux en termes de coûts, de budget et de niveau de qualité	Mesure de succès : Satisfaction client par la livraison de valeur ajoutée

TABLE 2.1 – Tableau comparatif entre les méthodes traditionnelles et agiles

Après une étude comparative entre ces deux grandes familles de méthodes, on a décidé d'adopter une gestion de développement Agile car c'est elle qui répond parfaitement aux besoins du projet mais il reste encore à choisir parmi les méthodes agiles la méthodologie la plus adaptée à son projet. En effet les méthodes Agiles disponibles sont nombreuses et peuvent être source de confusion. Les méthodes Agiles les plus populaires en usage aujourd'hui sont :

- l'eXtreme Programming (XP) ;
- Scrum ;
- Feature Driven Development (FDD) ;
- Lean Software Development ;
- Kanban ;
- Agile Unified Process (Agile UP ou AUP) ;
- Crystal ;

— Dynamic Systems Development Method (DSDM).

Dans le tableau suivant, on fera une étude comparative entre deux de ces méthodologies à savoir Scrum et Kanban.

Thèmes	Scrum	Kanban
Planification	Au début de chaque sprint	Kanban board, Flux continu
Estimation de l'effort	Au début de chaque sprint	Optionnel
Changement de périmètre	Doit attendre le sprint suivant	Selon besoin
Rôles	Scrum master (SM) / product owner (PO) / développeur (Dev)	Aucun rôle imposé
Méthodologie de livraison	À la fin de chaque sprint	Livraison continue
Top 3 bénéfices	<ul style="list-style-type: none"> — Productivité, — Scalabilité, — Engagement des équipes 	<ul style="list-style-type: none"> — Mise en place rapide sans changement des processus existants, — Pilotage visuel, — Gestion des files d'attentes Flux.

TABLE 2.2 – Tableau comparatif entre Scrum et Kanban

2.3.2 Méthodologie adoptée : Kanban

Parmi ces méthodologies, on a choisi Kanban car c'est une méthode de développement agile mais aussi de gestion du changement qui permet d'améliorer sa façon de travailler en s'appuyant sur la visualisation du flux de travail et dont les ressources sont régulièrement actualisées. En plus, elle répond parfaitement à nos besoins.

Kanban, est une méthode agile de gestion de projet particulièrement bien adaptée aux projets de développement informatique.

Le projet ainsi que les fiches Kanban ont été conçues avec un outil de conduite de projet à savoir Jira[10] (voir Annexe B 55).

Elle se caractérise par ces quatres (4) principes ainsi que cinq (5) pratiques.[11]

2.3.2.1 Les principes de Kanban

Commencez par ce que q'on fait actuellement : la méthode Kanban utilise les processus déjà en place et encourage une amélioration de ces processus.

Acceptez d'appliquer des changements progressifs : les collaborateurs sont généralement résistants aux changements trop brutaux et radicaux. L'équipe doit accepter d'améliorer le système en place par des changements progressifs.

Respectez le processus actuel, les rôles, les responsabilités et les titres :

Afin de faciliter les changements à venir, il faut éliminer la crainte du changement, notamment en maintenant et en respectant les rôles, les responsabilités et les titres professionnels actuels de chacun.

Leadership à tous les niveaux : collaborateurs ou cadres supérieurs, tous les actes de leadership au sein de l'organisation doivent être encouragés.

TÂCHE 1 ISSUE	A FAIRE 1 ISSUE	EN COURS 2 ISSUES	A TESTER 2 ISSUES	FINI 3 ISSUES ✓
Gestion des paiements + Créer un ticket	Gestion des demandes de propriétés + BNBPRO-4	Gestion des paramétrages Matériel TypeOffre Equipement TypePropriété ModèleContrat + BNBPRO-9	Gestion des contrats + BNBPRO-5 Gestion des tâches + BNBPRO-7	Gestion des utilisateurs Signup AddCollab Signin Authentification + BNBPRO-1
		Gestion des dépenses + BNBPRO-6		Gestion des clients + BNBPRO-2
				Gestion de la propriétés + BNBPRO-3

FIGURE 2.3 – Tableau Kanban à la date du 25/05/2021

2.3.2.2 Les pratiques de Kanban

La méthodologies Kanban dispose de bonnes pratiques qui facilitent le développement des applications logicielles. Voir le détail en Annexe A 54.

2.3.3 Chronogramme

Le tableau suivant montre le planning de ce projet , on l'a réalisé avant de commencer à travailler sur le projet pendant la phase de l'élaboration du cahier

des charges.

Il présente l'enchaînement des tâches effectuées tout le long de ce stage :

Tâche	Date début	Date fin	Détails
Documentation	08/03/2021	01/05/2021	Prise en main des différentes TECHNOLOGIES qu'on a utilisées pour la réussite du projet
Elaboration des fiches Kanban	08/03/2021	13/04/2021	Planification
Tâche 1 « Gestion des utilisateurs »	08/03/2021	20/03/2021	gestion des connexions utilisateurs, ajout de nouveau collaborateurs, etc...
Tâche 2 « Gestion des clients »	15/03/2021	28/03/2021	ajout, modification, suppression, recherche et affichage des clients.
Tâche 3 « Gestion de la propriété »	22/03/2021	11/04/2021	opération SCRUD (search, create, read, update, delete)
Tâche 4 « Gestion des demandes de propriétés »	05/04/2021	17/04/2021	faire des demandes ; répondre à des demandes.
Tâche 5 « Gestion des dépenses »	03/05/2021	16/05/2021	consulter les dépenses ; suivre des dépenses.
Tâche 6 « Gestion des tâches »	02/05/2021	24/05/2021	ajouter des tâches ; les consulter ; les effectuer ;
Tâche 7 « Gestion des contrats »	18/04/2021	09/05/2021	créer des contrats et des modèles de contrats ;
Tâche 8 « Gestion des paiements »	16/05/2021	29/05/2021	consulter les paiements ; les enregistrer.
Tâche 9 « Gestion des paramétrages »	23/05/2021	05/06/2021	Gestion des types propriétés, d'offres, de matériels, d'équipement...

TABLE 2.3 – Chronogramme des tâches

2.3.4 Diagramme de Gantt

Le diagramme de GANTT est un outil permettant de modéliser la planification de tâches nécessaires à la réalisation du projet, il visualise dans le temps les diverses tâches liées composant le projet en fournissant une représentation graphique de son avancement. La figure ci-dessous montre le diagramme de Gantt

de ce projet. Il a été réalisé avec un outil d'édition de diagramme en ligne appelé VisualParadigm[12].

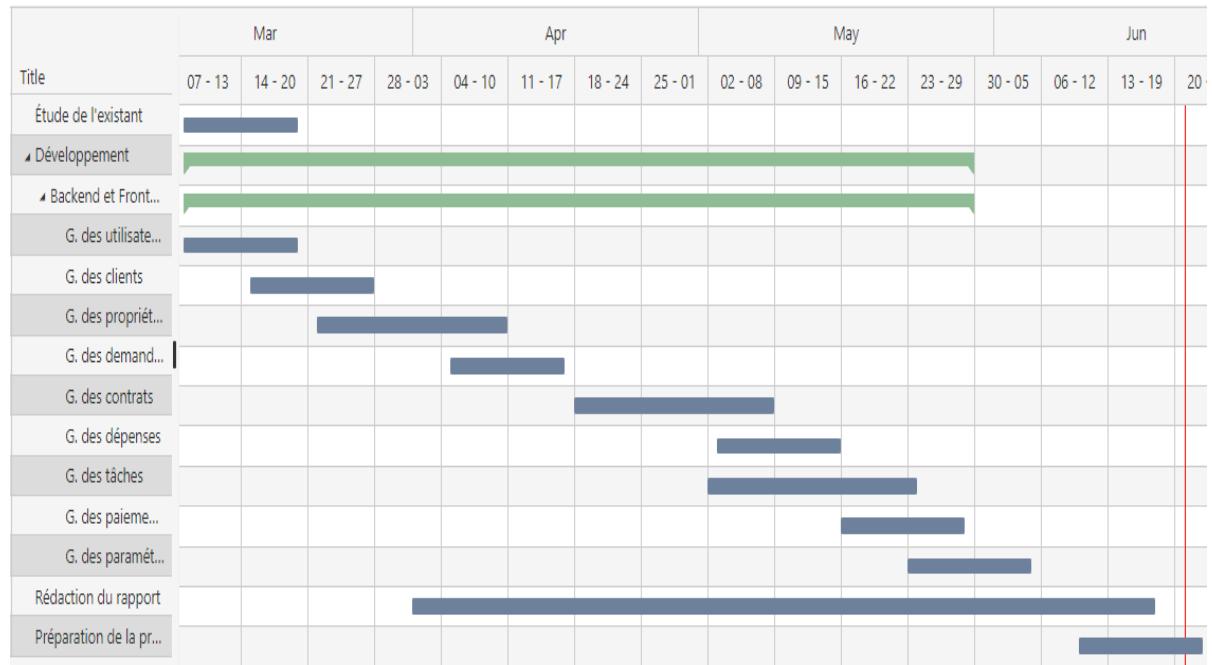


FIGURE 2.4 – Diagramme de GANTT

2.4 Conclusion

Dans ce premier chapitre, on a fait une étude de l'existant et les critiques de celui-ci, sans oublier la présentation de l'organisme d'accueil ainsi que le projet à réaliser. Il a permis de mieux connaître et comprendre les besoins fonctionnels et non fonctionnels du système.

On va entamer maintenant la phase de conception.

Chapitre 3

ÉTUDE CONCEPTUELLE

3.1 Introduction

L'étape de conception constitue un pont entre la définition de besoins et l'implémentation. La conception décrit le fonctionnement du système par une étude générale basée sur la spécification des besoins en premier lieu, puis une étude détaillée du projet par plusieurs approches.

3.2 Choix de la méthode de conception

Pour atteindre les objectifs préfixés, il faut planifier et organiser les tâches à faire en s'alignant à une méthode de conception. Tout au long de ce projet, on va suivre le processus RUP (Rational Unified Process) comme méthode de conception orientée objet.

En effet, ce processus est bien adapté à UML[13] qui est l'acronyme anglais pour « Unified Modeling Language ». On le traduit par « Langage de modélisation unifié ». La notation UML est un langage visuel constitué d'un ensemble de schémas, appelés diagrammes, qui donnent chacun une vision différente du projet à traiter. UML fournit donc des diagrammes pour représenter l'application à développer. [14]

3.3 Modélisation des besoins fonctionnels

3.3.1 Diagramme des cas d'utilisation

Le diagramme des cas d'utilisation décrit les utilisations requises d'un système, ou ce qu'un système est supposé faire. Les principaux concepts de ce diagramme sont les acteurs, cas d'utilisation et sujets. Un sujet représente un système avec

lequel les acteurs et autres sujets interagissent[15].

Le diagramme du cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel.

Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système.

La relation « include » est utile car tous les utilisateurs doivent obligatoirement s'identifier afin d'accéder à l'application.

La figure suivante présente le diagramme des cas d'utilisation général :

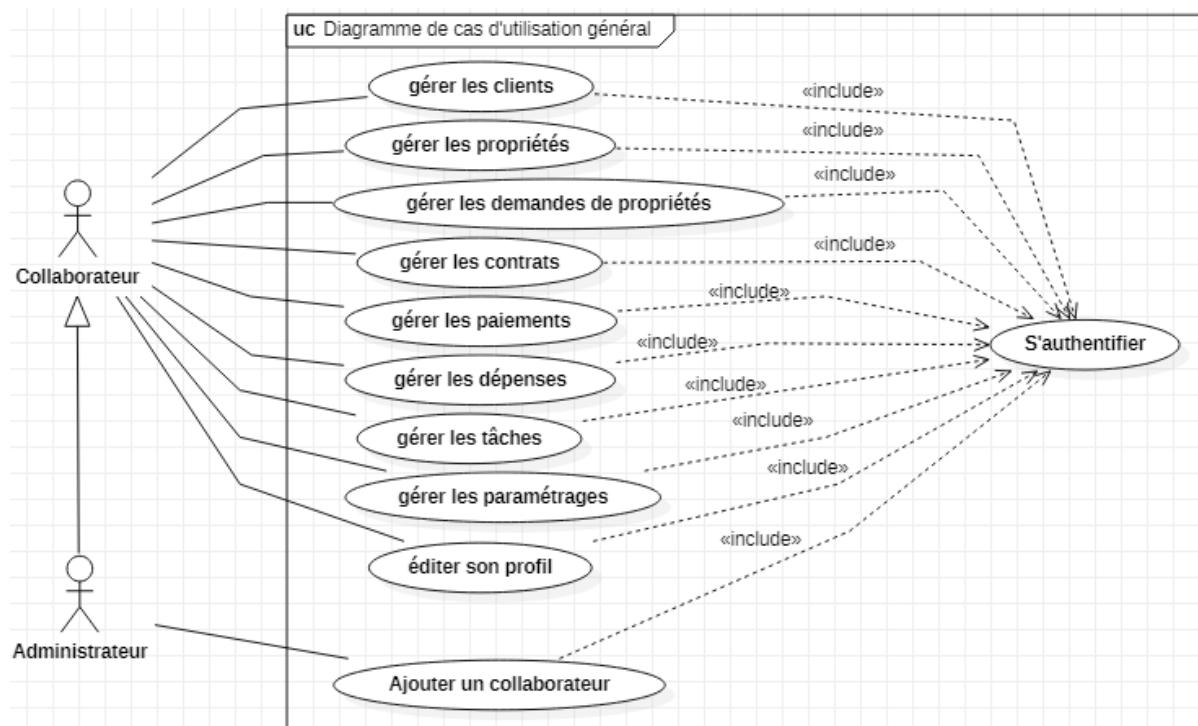


FIGURE 3.1 – Diagramme des cas d'utilisation général

3.3.2 Description détaillée des cas d'utilisation

En se référant au chronogramme présenté au niveau du tableau 2.3 à la page 19, on présente dans ce qui suit les descriptions de quelques cas d'utilisations exprimant le cahier de charge du projet.

3.3.2.1 Description du cas d'utilisation « S'authentifier » relatif à la tâche 1 « Gestion des utilisateurs »

Nom du cas d'utilisation	S'authentifier
Acteurs	administrateur et collaborateur
Préconditions	-le système fonctionne -l'utilisateur saisit son email et son mot de passe
Enchaînement nominal	1- Le système vérifie si les champs ne sont pas vides 2- Il vérifie ensuite si les informations sont valides 3- Le système redirige l'acteur vers son espace selon son rôle
Enchaînement alternatif	-En (1) : message d'erreur « un champ est vide » ; -En (2) : message d'erreur « émail ou mot de passe incorrect »
Exceptions	Le système affiche un message d'erreur dans le cas où le serveur ne fonctionne pas
Postconditions	Ouverture de l'espace de l'acteur

TABLE 3.1 – Description du cas d'utilisation « S'authentifier »

3.3.2.2 Description du cas d'utilisation « Ajouter un collaborateur » relatif à la tâche 1 « Gestion des utilisateurs »

Nom du cas d'utilisation	Ajouter un collaborateur
Acteurs	administrateur
Préconditions	- Le système fonctionne - L'acteur est authentifié
Enchaînement nominal	1- L'acteur remplit le formulaire d'ajout 2- Le système vérifie si les champs ne sont pas vides 3- Le système vérifie si le format des données sont correctes 4- Le système valide l'ajout et affiche un message de succès
Enchaînement alternatif	-En (2) : message d'erreur « ce champ est obligatoire » -En (3) : message d'erreur « Veuillez respecter le format des données »
Exceptions	Le système affiche un message d'erreur dans le cas où le serveur ne fonctionne pas
Postconditions	Le collaborateur est ajouté avec succès

TABLE 3.2 – Description du cas d'utilisation « ajouter un collaborateur »

3.3.2.3 Description du cas d'utilisation « Gérer les clients » relatif à la tâche 2 « Gestion des clients »

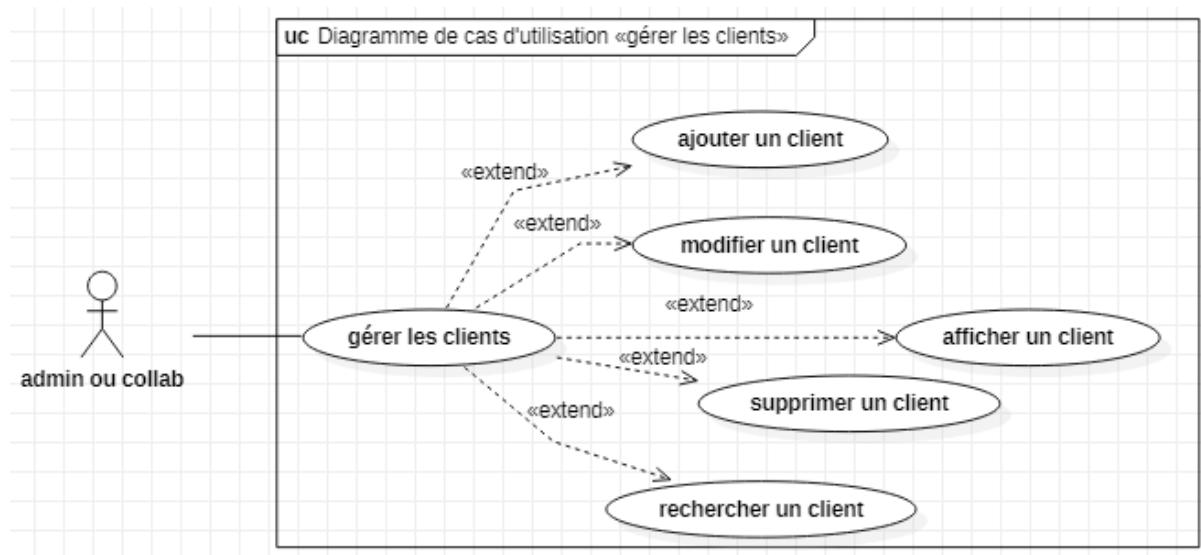


FIGURE 3.2 – Diagramme du cas d'utilisation « *gérer les clients* »

Nom du cas d'utilisation	Ajouter un client
Acteurs	administrateur ou collaborateur
Préconditions	- Le système fonctionne - L'acteur est authentifié
Enchaînement nominal	1- L'acteur remplit le formulaire d'ajout 2- Le système vérifie si les champs ne sont pas vides 3- Le système vérifie si le format des données sont correctes 4- Le système valide l'ajout et affiche un message de succès
Enchaînement alternatif	-En (2) : message d'erreur « ce champ est obligatoire » -En (3) : message d'erreur « Veuillez respecter le format des données »
Exceptions	Le système affiche un message d'erreur dans le cas où le serveur ne fonctionne pas
Postconditions	Le client est ajouté avec succès

TABLE 3.3 – Description du cas d'utilisation « *Ajouter un client* »

Nom du cas d'utilisation	Modifier un client
Acteurs	administrateur ou collaborateur
Préconditions	- Le système fonctionne - L'acteur est authentifié
Enchaînement nominal	1- L'acteur choisit le client à modifier 2- Le système affiche le formulaire de modification 3- L'acteur remplit le formulaire de modification 4- Le système vérifie si les champs ne sont pas vides 5- Le système vérifie si le format des données sont correctes 6- Le système valide la modification et affiche un message de succès
Enchaînement alternatif	-En (4) : message d'erreur « ce champ est obligatoire » -En (5) : message d'erreur « Veuillez respecter le format des données »
Exceptions	Le système affiche un message d'erreur dans le cas où le serveur ne fonctionne pas
Postconditions	Le client est modifié avec succès

 TABLE 3.4 – Description du cas d'utilisation « *modifier un client* »

Nom du cas d'utilisation	Supprimer un client
Acteurs	administrateur ou collaborateur
Préconditions	- Le système fonctionne - L'acteur est authentifié
Enchaînement nominal	1- L'acteur choisit le client à supprimer 2- Le système demande la confirmation de suppression 3- L'acteur confirme la suppression 4- Le système valide la suppression et affiche un message de succès
Exceptions	Le système affiche un message d'erreur dans le cas où le serveur ne fonctionne pas
Postconditions	Le client est supprimé avec succès

 TABLE 3.5 – Description du cas d'utilisation « *supprimer un client* »

Nom du cas d'utilisation	Rechercher un client et l'afficher
Acteurs	administrateur ou collaborateur
Préconditions	- Le système fonctionne - L'acteur est authentifié
Enchaînement nominal	1- L'acteur saisit le client à rechercher 2- Le système lance la demande de recherche 3- L'acteur voit le résultat de la recherche 4- L'acteur clique et affiche les informations du client recherché
Enchaînement alternatif	-En (3) : message d'erreur « client non trouvé »
Exceptions	Le système affiche un message d'erreur dans le cas où le serveur ne fonctionne pas
Postconditions	Le client est trouvé

 TABLE 3.6 – Description du cas d'utilisation « *rechercher un client et l'afficher* »

3.4 Modélisation dynamique

3.4.1 Diagrammes de séquences

Un diagramme de séquence est un diagramme d'interaction dont le but est de décrire comment les objets collaborent au cours du temps et quelles responsabilités ils assumment. Il décrit un scénario d'un cas d'utilisation.[16]

Avec les intéressants ajouts au diagramme de séquences apportés par UML 2, en particulier les cadres d'interactions (avec les opérateurs loop, opt et alt par exemple), ainsi que la possibilité de référencer une interaction décrite par ailleurs, le diagramme de séquence système semble constituer une excellente solution.

Dans le cas des applications informatiques, il existe des stéréotypes d'objets qui permettent de séparer les préoccupations de l'IHM, l'application et les données.



Boundary : est une interface graphique ou une autre interface Homme/Machine (écran tactile, PocketPC, ...).

Entity : est une classe qui sert à modéliser des informations durables et souvent persistantes.

Control : est une classe utilisée pour représenter la coordination, l'enchaînement et le contrôle d'autres objets.

Le schéma ci-dessous explique le fonctionnement de l'architecture MVC (Model-View-Controller)

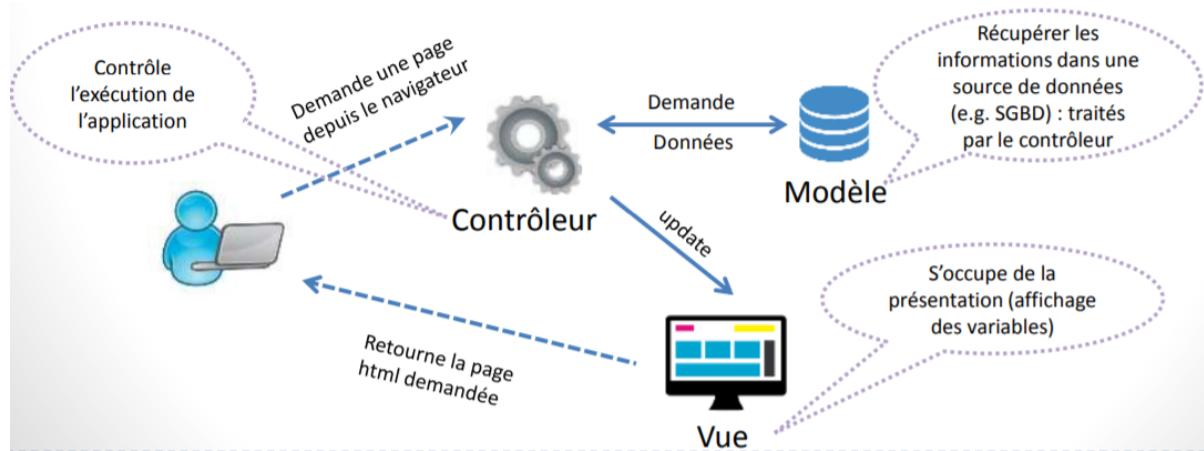


FIGURE 3.3 – Fonctionnement de l'architecture MVC

3.4.1.1 Diagramme de séquence d'authentification relatif à la tâche 1 « Gestion des utilisateurs »

L'utilisateur se connecte à l'application via un compte local. Il saisit son adresse email et son mot de passe, le système vérifie les données saisies et envoie une requête d'authentification à la base de données. Si les données correspondent à un compte valide, la connexion sera établie sinon un message d'erreur sera affiché.

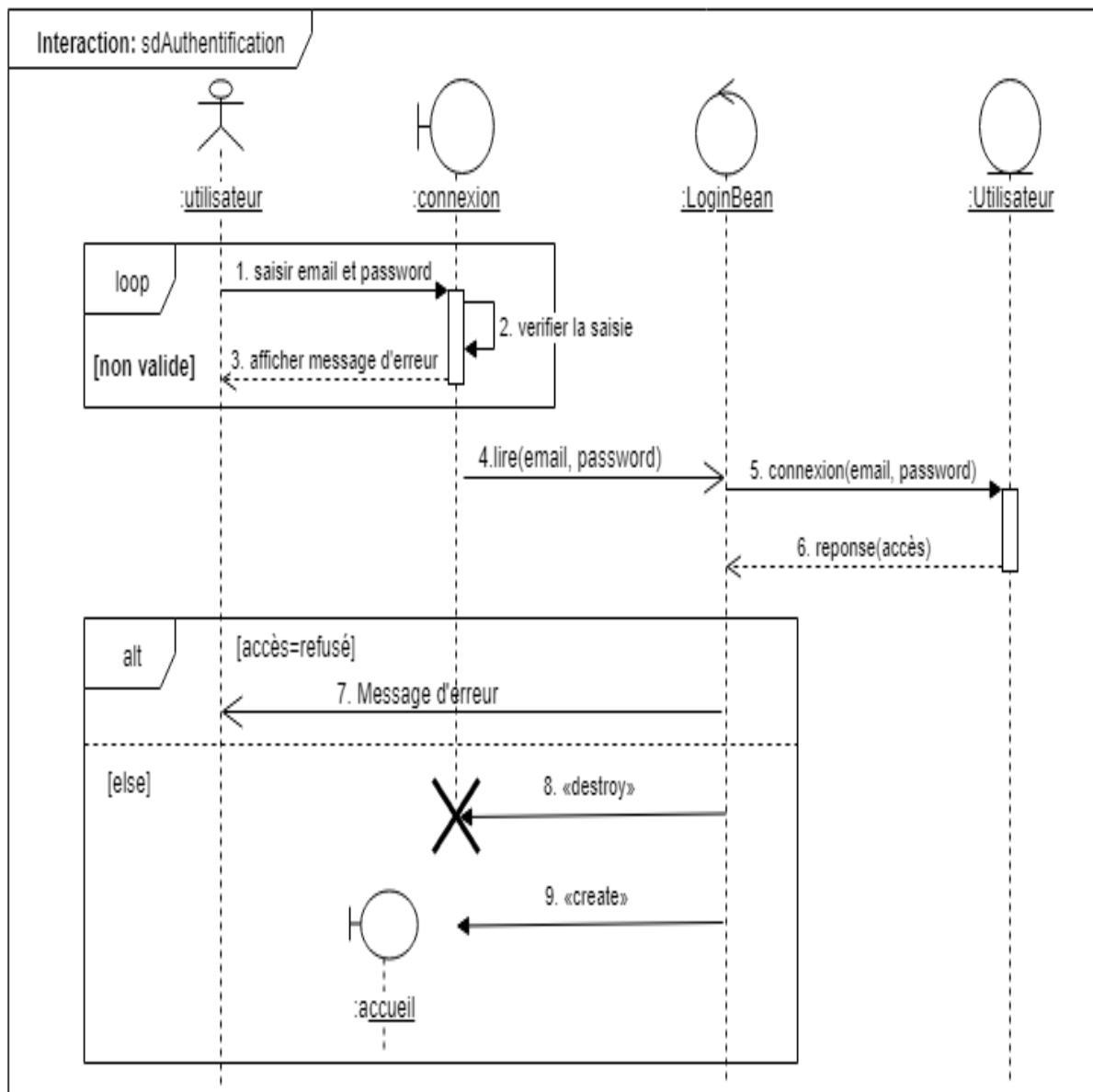


FIGURE 3.4 – Diagramme de séquence d’Authentification

3.4.1.2 Diagramme de séquence de « gérer les dépenses » relatif à la tâche 5 « Gestion des dépenses »

L’administrateur s’authentifie d’abord, puis il accède à la page d’administration du système, ensuite il choisit la gestion des dépenses. Sur cette page il a la possibilité d’ajouter, modifier ou supprimer une dépense.

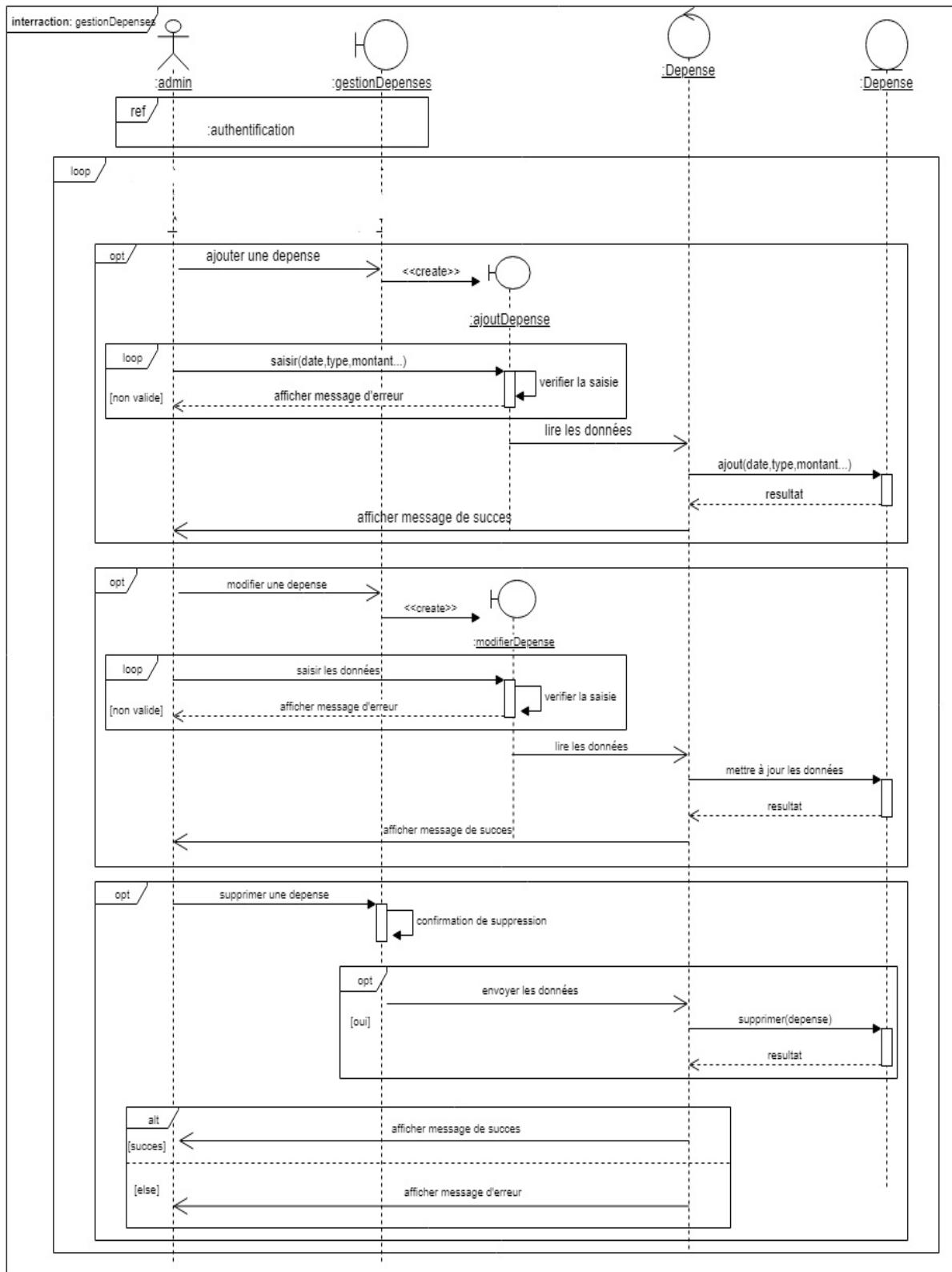


FIGURE 3.5 – Diagramme de séquence « gérer les dépenses »

3.4.2 Diagrammes d'activité

Les diagrammes d'activité sont la représentation graphique des interactions entre acteurs et le système selon un ordre logique dans la formulation UML. On montre ces interactions dans le cadre d'un scénario d'un diagramme de cas d'utilisation. Le but est de décrire comment se déroulent les actions entre les acteurs ou objets.

3.4.2.1 Diagramme d'activité du cas d'utilisation « Incription » relatif à la tâche 1 « Gestion des utilisateurs »

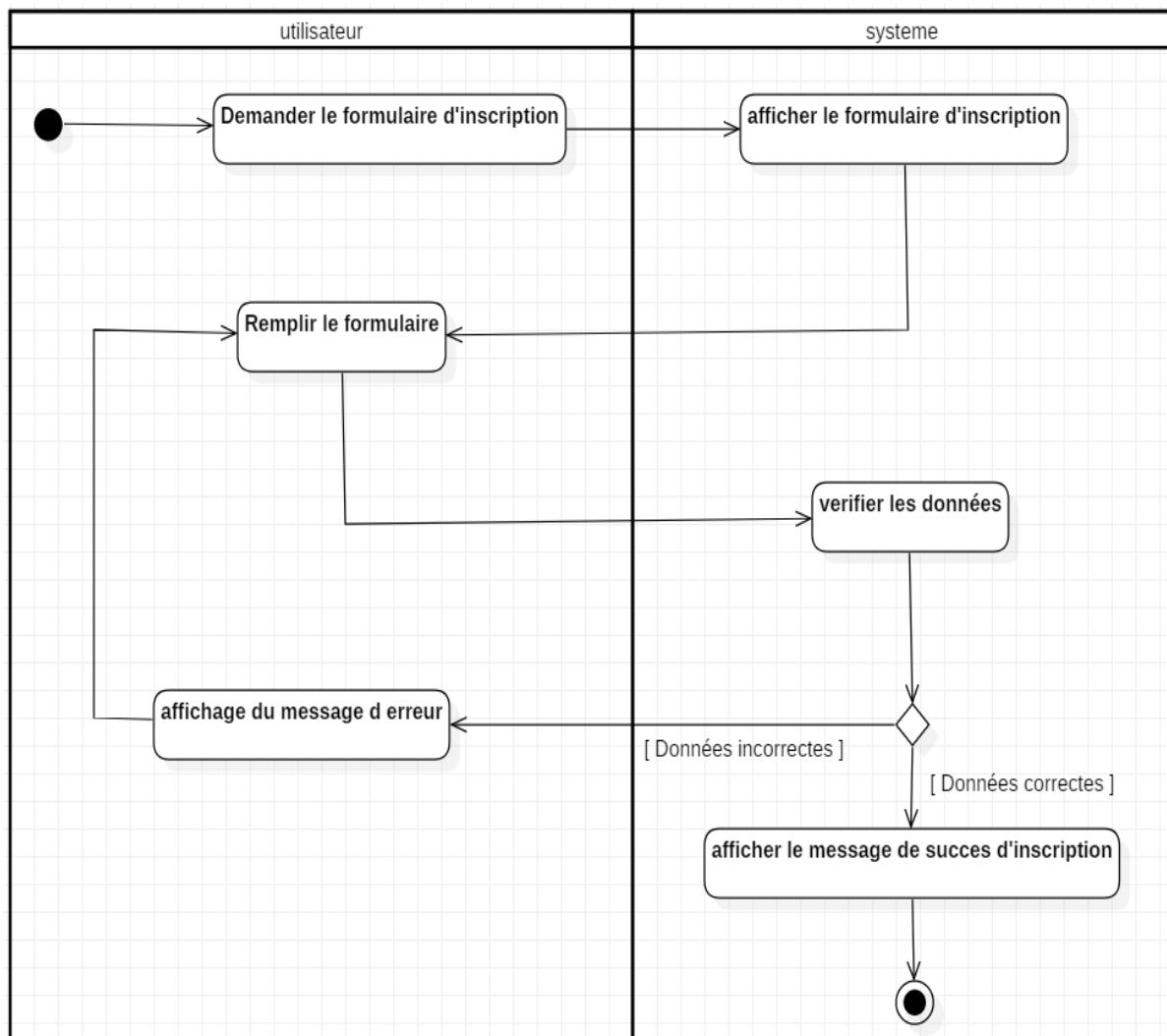


FIGURE 3.6 – Diagramme de d'activite « inscription »

3.4.2.2 Diagramme d'activité du cas d'utilisation « Ajouter une dépense » relatif à la tâche 5 « Gestion des dépenses »

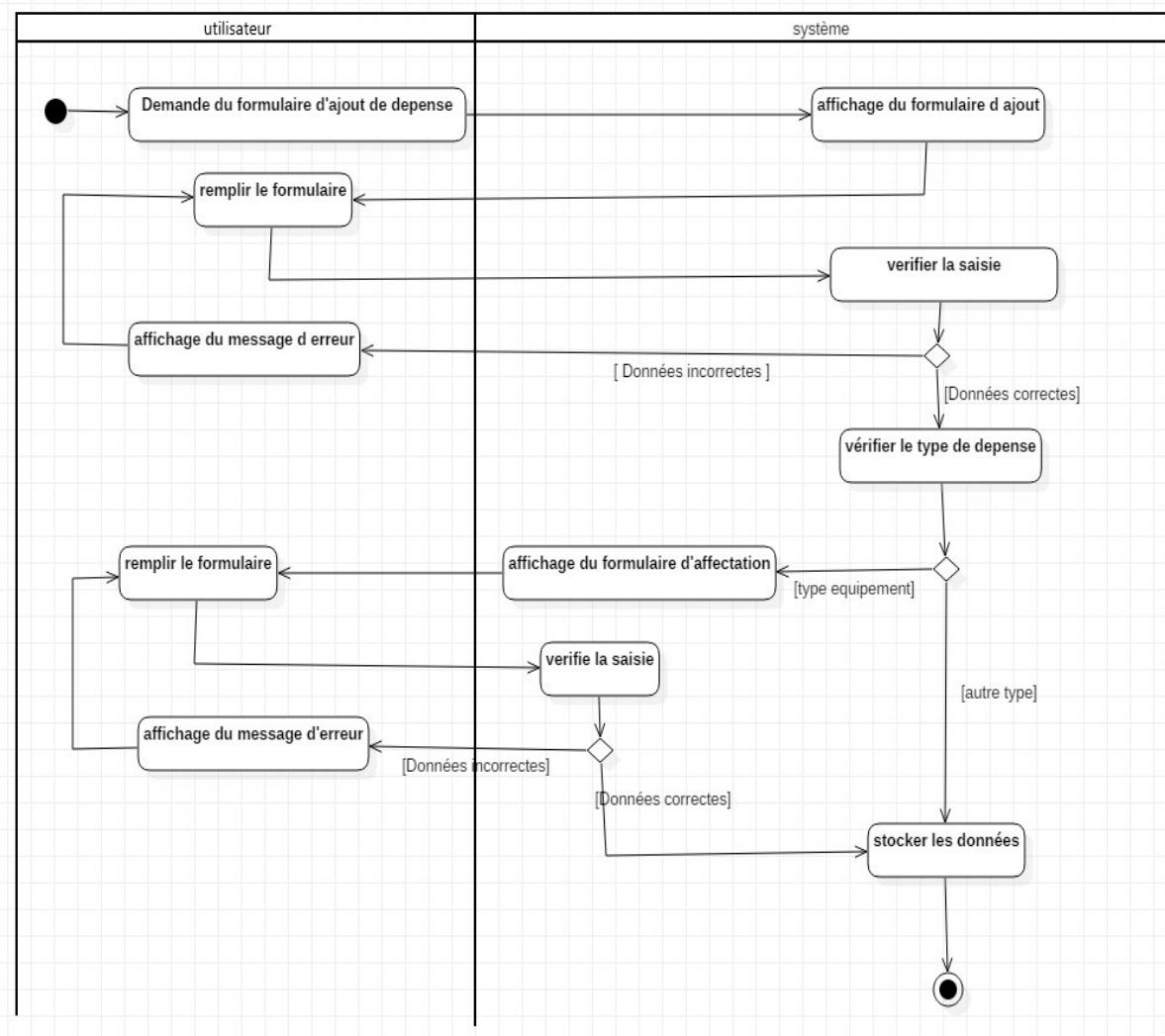
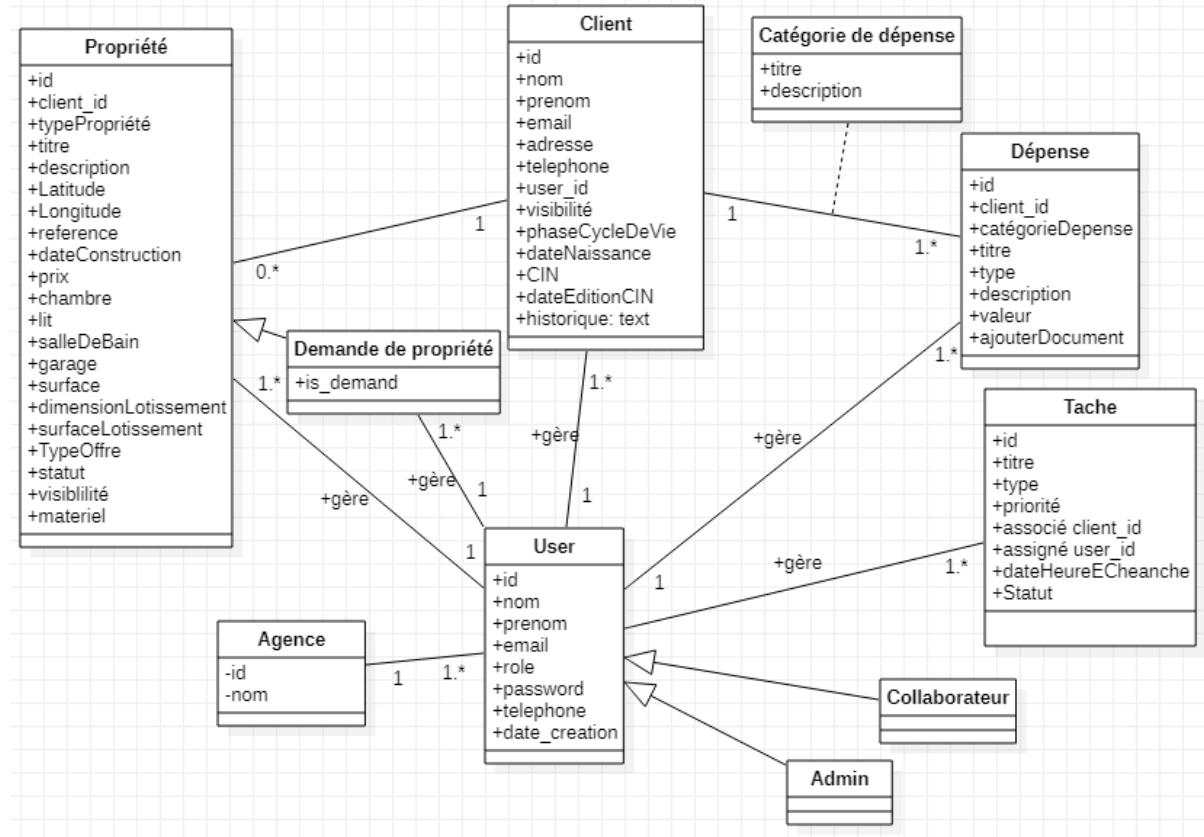


FIGURE 3.7 – Diagramme de d'activité « ajouter une dépense »

3.5 Modélisation statique

3.5.1 Diagramme de classes

Le diagramme de classe constitue un élément très important de la modélisation : Il permet de définir quelles seront les composantes du système final. On constate souvent qu'un diagramme de classe proprement réalisé permet de structurer le travail de développement de manière très efficace ;


 FIGURE 3.8 – Diagramme de classes relatif à la tâche 6 « *Gestion des tâches* »

3.5.2 Diagramme de déploiement

Dans UML, les diagrammes de déploiement modélisent l'architecture physique d'un système. Les diagrammes de déploiement affichent les relations entre les composants logiciels et matériels du système, d'une part, et la distribution physique du traitement, d'autre part. Les diagrammes de déploiement, qu'on prépare généralement pendant la phase d'implémentation du développement, présentent la disposition physique noeud et les composants et autres éléments que les artefacts implémentent. Les noeuds représentent des périphériques matériels tels que des ordinateurs, des détecteurs et des imprimantes, ainsi que d'autres périphériques qui prennent en charge l'environnement d'exécution d'un système. Les chemins de communication et les relations de déploiement modélisent les connexions dans le système. Le diagramme de déploiement suivant met en évidence les différents niveaux de l'application.

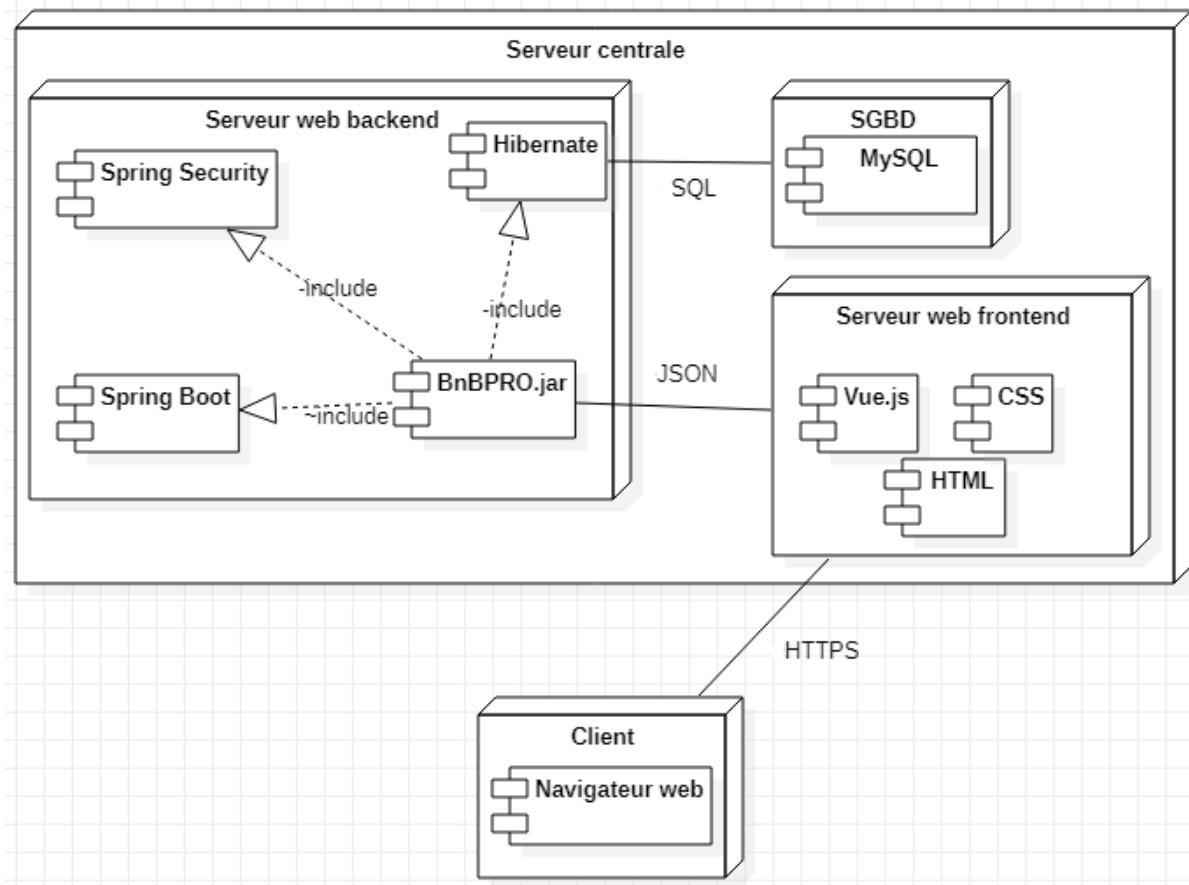


FIGURE 3.9 – Diagramme de déploiement général

3.6 Conclusion

Ce chapitre a permis de donner une image claire et nette du système. On a spécifié les cas d'utilisation du système, les diagrammes d'activité et de séquence, le diagramme de classe et le diagramme de déploiement spécifiant l'infrastructure physique du système.

Chapitre 4

DÉVELOPPEMENT DE L'APPLICATION

4.1 Introduction

Après avoir étudié l'existant et le critiquer, on a exprimé les besoins fonctionnels et non fonctionnels de la solution, ensuite on a fait une conception de celle-ci. Il ne reste plus qu'à l'implémenter.

Dans ce chapitre, on présentera les techniques utilisées, l'environnement matériel et le logiciel de développement ainsi que les étapes de réalisation de cette solution en se basant sur quelques captures d'écrans.

4.2 Choix technologiques

Notre projet se compose de deux parties : la partie backend[17] pour la création des services¹ et la partie frontend pour les interfaces. Pour développer la partie backend, on a utilisé les frameworks Spring, Spring Boot et Spring Security, et pour l'application web, on a utilisé essentiellement le framework Vue.js .

Tous les Framework utilisés sont open source et totalement gratuit, ce qui un plus pour le projet. En effet la société McZEN TECHNOLOGIES n'utilise que des Frameworks gratuits. En outre les différentes documentation de ceux-ci sont complètent et régulièrement mises à jour lors de la diffusion de chaque nouvelle version. C'est un avantage majeur pour moi en tant que débutant dans l'utilisation de ces technologies.

1. Service, une fonctionnalité ou partie de fonctionnalité mise à disposition par un composant logiciel pour assurer une tâche particulière.

4.2.1 Notion de Microservices

C'est quoi un Microservice ?	L'architecture Microservices propose une solution en principe simple : découper une application en petits services, appelés Microservices, parfaitement autonomes. Chaque microservice répond à un objectif métier spécifique et communique avec les autres en utilisant des API indépendamment du langage de programmation.
C'est quoi API REST ?	Une API (Application Programming Interface) de type REST (REpresentational State Transfer), ou « RESTful », est une interface de programmation d'application qui fait appel à des requêtes HTTP pour obtenir (GET), placer (PUT), publier (POST) et supprimer (DELETE) des données.
Pourquoi API REST ?	On a déployé une API REST pour assurer les différentes interactions entre l'application web et le serveur de base de données.

TABLE 4.1 – Descriptif de la notion de Microservices

Pour tester le fonctionnement des API, on s'est servi de l'outil Postman (voir le tableau 4.3.2.2 à la page 37) pour certifier leur bon fonctionnement, avant de démarrer le codage dans la partie frontend.

4.3 Environnement de travail

En informatique, un environnement désigne, pour une application, l'ensemble des matériels et des logiciels système, dont le système d'exploitation, sur lesquels sont exécutés les programmes de l'application.[18]

4.3.1 Environnement matériel

L'environnement matériel est l'ensemble du matériel, sur lequel le code de l'application est développé et exécuté.

Pour développer ce projet, on a utilisé un ordinateur de la marque "acer" ayant les caractéristiques suivantes :

- **Système d'exploitation** : Windows 10 x64.
- **Processeur** : Intel Core i5 2410M.
- **RAM** : 4Go
- **Disque dur** : 450Go SSD

4.3.2 Environnement logiciel

Dans cette section, on fait la description de tous les logiciels qu'on a utilisés pour la réalisation de l'application.

4.3.2.1 Environnements de développement

Un environnement de développement communément appelé IDE (Integrated Development Environment) est un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels, des sites web, etc.. Il comporte un éditeur de texte destiné à la programmation, des fonctionnalités telles que le débogage et tant d'autres. Certains de ces environnements sont dédiés à un langage de programmation en particulier.

Spring Tool Suite	Spring Tool Suite est un IDE pour développer des applications Spring. Il s'agit d'un environnement de développement basé sur Eclipse. Il fournit un environnement prêt à l'emploi pour implémenter, exécuter, déployer et déboguer l'application. Il valide l'application et fournit des correctifs rapides pour les applications. [19]
Atom	Atom est un éditeur code extensible développé par GitHub pour macOS, GNU/Linux et Windows. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires. [20]
TeXnicCenter	TeXnicCenter se présente sous la forme d'un IDE pour le langage LaTeX sous Windows. Il permet à l'utilisateur d'écrire un rapport ou un document LaTeX et de le compiler dans les formats PDF, DVI ou PS. Un menu permet un accès simple et rapide aux éléments usuels (formules, symboles, sections). On peut aussi insérer une bibliographie ainsi qu'un index. [21]

TABLE 4.2 – Descriptif des éditeurs utilisés

4.3.2.2 Logiciels divers

Postman	Postman est un outil de développement d'API (interface de programmation d'application) qui permet de créer, tester et modifier des API. Presque toutes les fonctionnalités dont un développeur pourrait avoir besoin sont encapsulées dans cet outil. Il a la capacité de faire divers types de requêtes HTTP (GET, POST, PUT, PATCH), d'enregistrer des environnements pour une utilisation ultérieure, de convertir l'API en code pour divers langages (comme JavaScript, Python). (Voir la démonstration au niveau de l'Annexe E à la page 64. [22])
MySQL Workbench	MySQL Workbench est un logiciel de gestion et d'administration de bases de données MySQL, permettant l'accès local (sans avoir à se connecter à un serveur externe) à des bases de données. Via une interface graphique intuitive, il permet, entre autres, de créer, modifier ou supprimer des tables, des comptes utilisateurs, et d'effectuer toutes les opérations inhérentes à la gestion d'une base de données. Pour ce faire, il doit être connecté à un serveur MySQL. [23]

TABLE 4.3 – Descriptif des logiciels utilisés

4.3.3 Frameworks de développement

Dans cette section, on fait la description de tous les Frameworks dont on s'est servi pour le backend et pour le frontend.

4.3.3.1 Frameworks Backend

Spring	
Description	Pourquoi utiliser Spring ?
Spring est un Framework open source pour construire et définir l'infrastructure d'une application Java, qui facilite le développement et les tests. Spring Framework fournit un modèle complet de programmation et de configuration pour les applications d'entreprise modernes basées sur Java, sur tout type de plate-forme de déploiement. [24]	Spring est composé de plusieurs modules disponibles sous forme de fichiers jar (Bibliothèques pour tous types de projets et pour tous les besoins de développeurs). Ainsi, on peut n'ajouter au projet que la partie que l'on souhaite utiliser. Cette richesse en fonctionnalités va permettre une évolution très rapide du futur système.
Hibernate	
Description	Pourquoi utiliser Hibernate ?
Hibernate est une solution de type ORM (Object Relational Mapping) qui permet de faciliter le développement de la couche persistance d'une application. Hibernate permet donc de représenter une base de données en objets Java et vice versa. Hibernate est très populaire notamment à cause de ses bonnes performances et de son ouverture à de nombreuses bases de données. [25]	Hibernate génère le code SQL nécessaire, ce qui rend l'application plus portable (s'adapte à la base de données). De plus on peut faire de vos classes métiers des classes persistantes sans ajout de code c'est à dire que le développeur pense en termes d'objet et pas en termes de lignes de tables et on peut interroger la base de données de plusieurs façons (Requête SQL, langage HQL...).

TABLE 4.4 – Tableau 1 : descriptif des Frameworks utilisés pour le Backend

Et ainsi que les Frameworks :

Spring Boot	C'est le Framework qui est le point d'entrée unique vers tous les projets de Spring que qu'on pourra donc utiliser dans vos applications. Il est particulièrement recommandé pour le développement d'API. Grâce à une configuration par défaut, une application Spring Boot permet de développer plus rapidement en se focalisant essentiellement sur le code métier. [26]
Spring Security	Spring Security est un Framework de sécurité léger qui fournit une authentification et un support d'autorisation afin de sécuriser les applications Spring. Très largement utilisée au sein de la communauté Spring, son avantage est sa portabilité c'est à dire qu'il ne dépend pas d'un serveur d'applications particulier, son utilisation est identique quel que soit le serveur utilisé. Cette portabilité est particulièrement importante si l'application développée doit pouvoir être utilisée à un grand nombre de clients possédant des systèmes hétérogènes. [27]

TABLE 4.5 – Tableau 2 : descriptif des Frameworks utilisés pour le Backend

4.3.3.2 Frameworks Frontend

Vue.js	
Description	Pourquoi utiliser Vue.js ?
Vue.js est un Framework JavaScript utilisé pour construire des interfaces utilisateur et des applications web monopages. Vue présente des fonctionnalités avancées requises pour les applications complexes telles que le routage, la gestion d'état et les outils de construction sont offertes par le biais de bibliothèques et de paquets officiellement maintenus [28]	Vue.js est le framework JS le plus léger du marché. Il utilise une approche orientée composants (.vue). Chaque composant est un bloc qui "se suffit à lui-même" disposant de son propre template, sa propre logique, voire son propre style. Cette approche permet un meilleur découpage d'une application et donc une meilleure évolutivité et maintenabilité dans le temps ainsi qu'une réutilisabilité ailleurs dans l'application.
Bootstrap	
Description	Bootstrap ?
Bootstrap est un framework qui contient un ensemble de codes HTML et CSS ainsi que des extensions JavaScript en option fonctionnant ensemble et qu'on va pouvoir utiliser pour créer des designs complexes de manière relativement simple. [29]	Bootstrap permet de gagner énormément de temps. On n'a pas besoin de coder toute la partie CSS. Il suffit de sélectionner les classes qu'on veut en codant la page HTML. Ensuite, grâce à l'enchaînement de classes prédéfinies, on structurera le code et obtiendra plus de cohérence dans celui-ci.

TABLE 4.6 – Descriptif des Frameworks utilisés pour le frontend

4.3.4 Outils divers

MySQL	MySQL est la base de données open source la plus populaire au monde. C'est un système de gestion de bases de données relationnelles SQL développé et supporté par Oracle. Il s'appuie sur le langage de requête structuré SQL (Structured Query Language). MySQL est également une base de données embarquée très populaire.
Jira	Jira est un outil de gestion de projet complet développé par Atlassian pour les équipes agiles. L'outil de gestion de projet permet de créer et de répartir plusieurs types de tickets (que cela soit une tâche à réaliser ou un bug à corriger) qui s'afficheront sur un tableau de bord. Les tâches à réaliser, celles en cours ainsi que les tâches àachever sont facilement identifiables par l'utilisateur.
StarUML	StarUML est un logiciel de modélisation UML. Étant simple d'utilisation, nécessitant peu de ressources système, ce logiciel constitue une excellente option pour une familiarisation à la modélisation et permet de concevoir une dizaine de types de diagrammes. [30]
GitLab	GitLab est une plateforme permettant d'héberger et de gérer des projets web de A à Z. Présentée comme la plateforme des développeurs modernes et une alternative de GitHub, elle offre la possibilité de gérer ses dépôts Git et ainsi de mieux appréhender la gestion des versions des codes sources. [31]

TABLE 4.7 – Descriptif de quelques outils utilisés

4.4 Les principales interfaces graphiques

Dans cette section, on va voir quelques interfaces tout en donnant une description de celles-ci.

4.4.1 Page d'authentification

Cette page est la première de l'application, elle permet aux différents utilisateurs de l'application d'accéder à leur espace.

Pour s'authentifier, l'utilisateur donne son email et son mot de passe et si les deux informations sont correctes alors il accède à son espace si non le système lui demande de saisir les informations correctes.

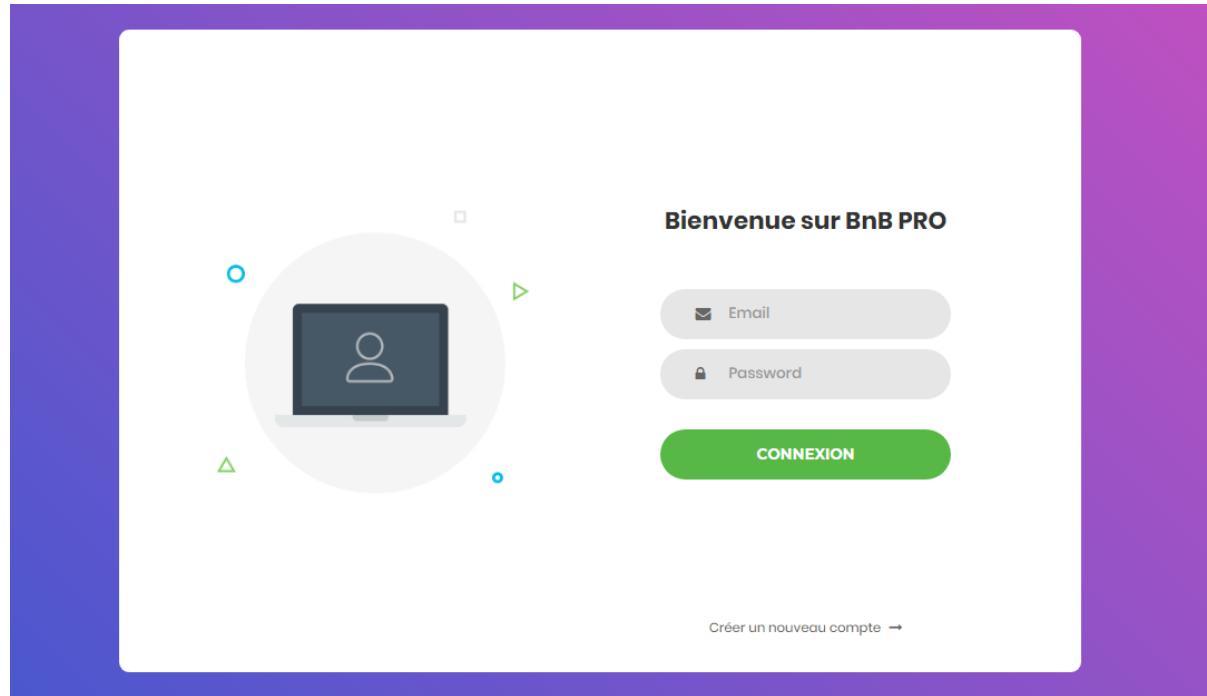


FIGURE 4.1 – Page d'authentification

4.4.2 Page de création de compte pour une agence

Cette page permet à l'internaute de créer un compte utilisateur et une agence en même temps et aussi d'accéder à son espace après validation.

Pour créer le compte, l'utilisateur donne les informations adéquats, s'ils sont correctes alors il accède à son espace sinon le système lui demande de saisir les informations correctes.

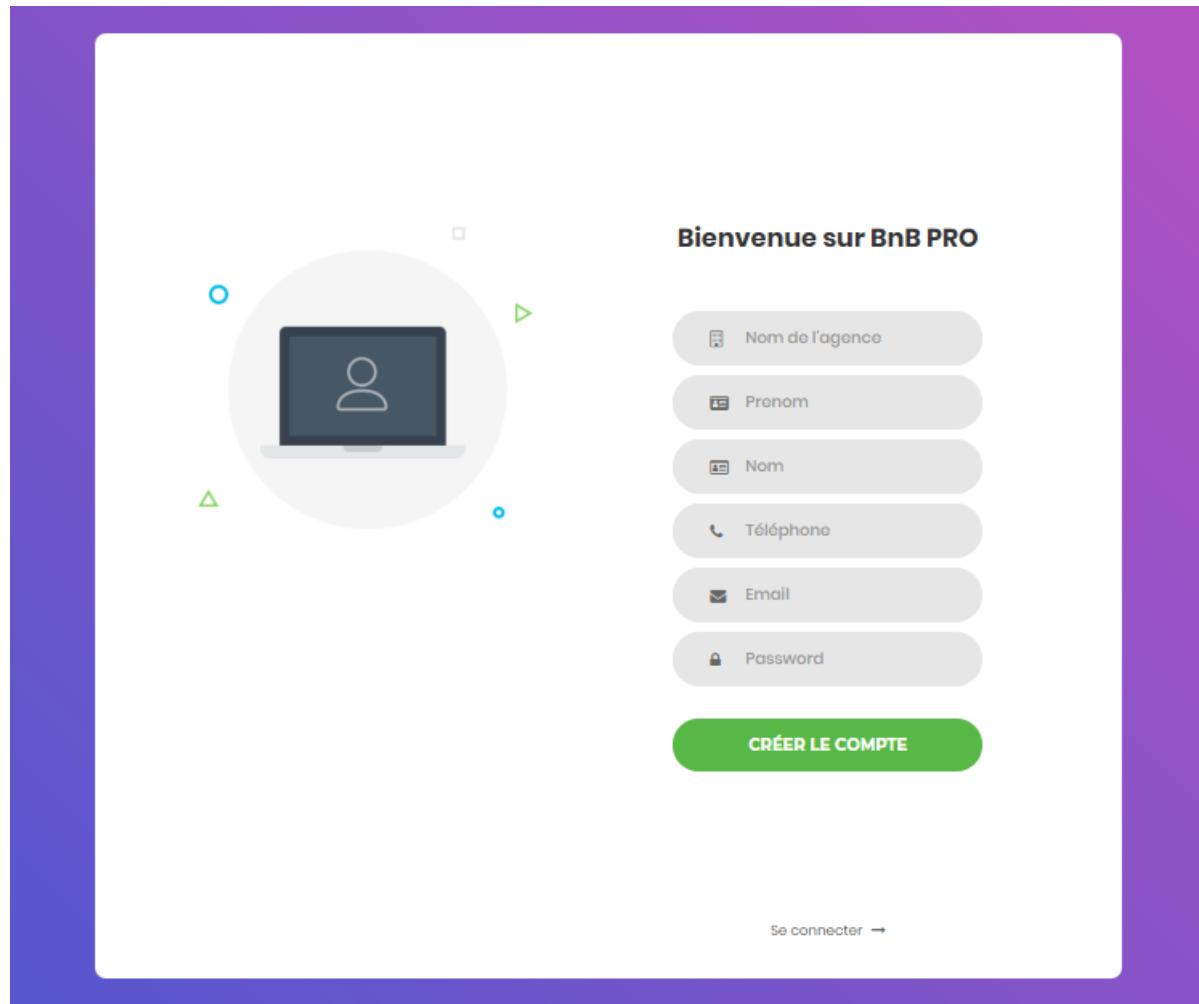


FIGURE 4.2 – Page de création de compte pour une agence

4.4.3 Gestion de la propriété

Après l'authentification, l'utilisateur accède à cette page via le menu de gauche qui la liste des propriétés. Sur cette page, il peut faire des recherches sur le titre, le client, le type, le type d'offre, la région, le prix ou la surface de la propriété.

On peut également à travers un clique sur les icônes soit consulter, éditer ou supprimer une propriété.

FIGURE 4.3 – Page de gestion de la propriété

On peut voir : Le bouton "Ajouter une nouvelle propriété" dans lequel on trouve notamment une carte pour la localisation de la propriété.

FIGURE 4.4 – Page de création de la propriété

On pourra grâce au menu de gauche accéder aux autres fonctionnalités de l'application telles que :

- La gestion des tâches ;
- La gestion des clients ;
- La gestion des propriétés ;

- La gestion des demandes de propriétés ;
- La gestion des contrats ;
- La gestion des dépenses ;
- La gestion des payements ;
- Et enfin la gestion des paramétrages ;

4.4.4 Crédit du modèle de contrat

Création d'un modèle de contrat personnalisé

Titre du modèle	Description du modèle
Contrat de vente	Saisissez la description du modèle
Paragraph ¶ A[¶] A[¶]I A[¶]I A[¶] A[¶] B I U S ≡ :	

Contrat de vente

entre

... (*nom et adresse à insérer*)

ci-après vendeur/venderesse

et

FIGURE 4.5 – Page de création du modèle de contrat

Ici on peut effectuer la création d'un modèle de contrat à l'aide de l'éditeur WYSIWYG Ckeditor.

4.5 Autres caractéristiques particulières de l'application

4.5.1 Persistance des données

En utilisant Hibernate qui profite de la souplesse qu'offre JPA², on facilite la persistannce et la recherche de données dans la base de données car il réalise lui-même la création des objets et les traitements de remplissage de ceux-ci en accédant à la base de données. La quantité de code ainsi épargnée est très importante d'autant que ce code est généralement fastidieux et redondant.

Pour fonctionner, Hibernate a besoin de plusieurs éléments :

- Une classe de type JavaBean qui encapsule les données d'une occurrence d'une table ;
- Une couche au-dessus de JPA que le Spring fournit à savoir Spring Data JPA [32] qui permet de définir des interfaces Repository (voir figure 2.2 à la page 15 en étendant avec CrudRepository ou JpaRepository.) ;
- Et enfin des propriétés de configuration notamment des informations concernant la connexion à la base de données ;

Voir l'Annexe D à la page 62 pour plus de détails.

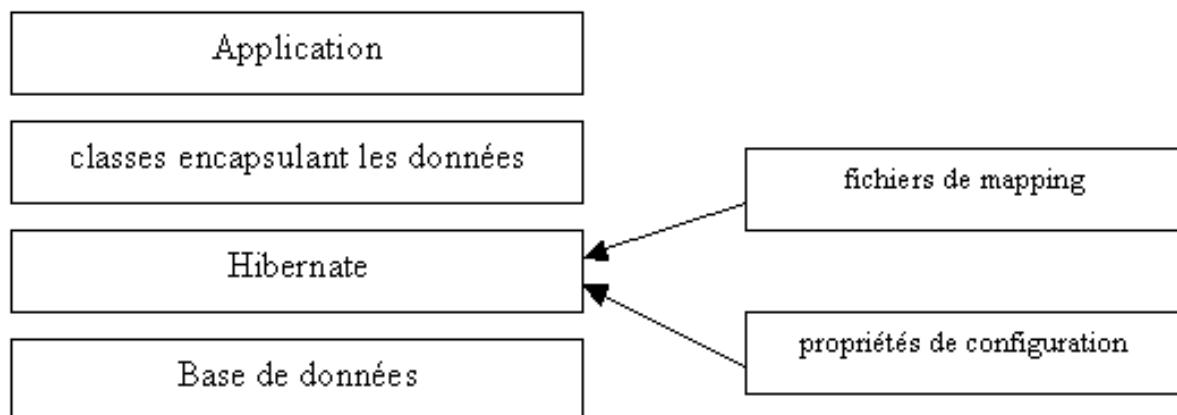


FIGURE 4.6 – Description du fonctionnement de la persistance des données

². La Java Persistence API est une API Java permettant d'organiser des données relationnelles dans des applications utilisant la plateforme Java

4.5.2 Sécurité

En outre l'utilisation de Spring Security pour la sécurisation de l'application, j'ai rajouté un mécanisme sécurisé d'échange d'informations appelé « JSON Web Token ».

Les JWT sont des jetons générés par un serveur lors de l'authentification d'un utilisateur sur une application Web, et qui sont ensuite transmis au client (navigateur). Ils seront renvoyés avec chaque requête HTTP au serveur, ce qui lui permettra d'identifier l'utilisateur.

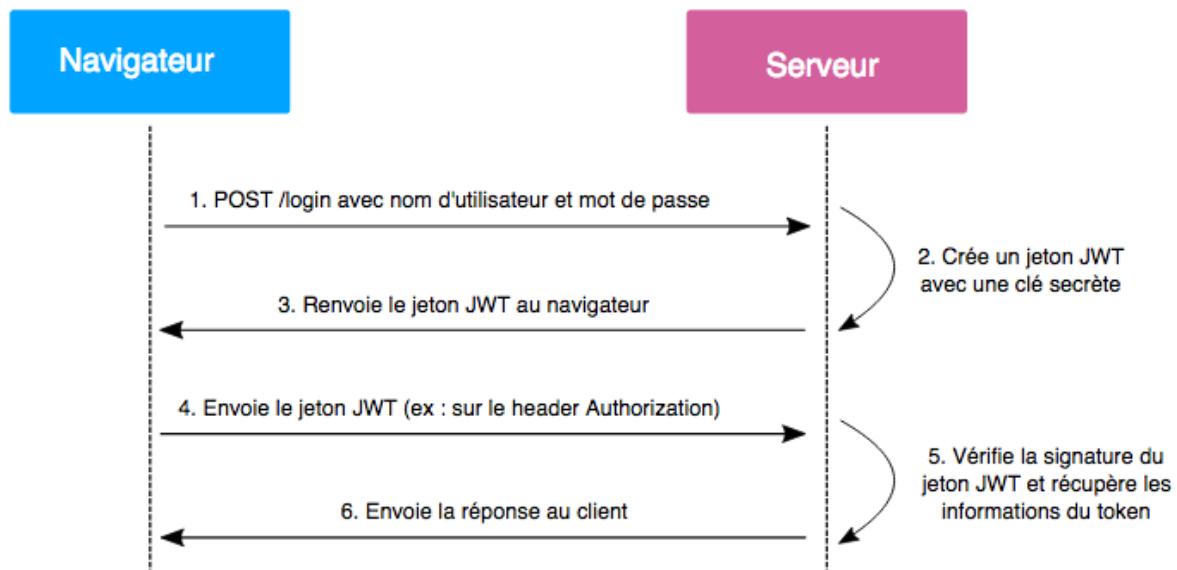


FIGURE 4.7 – Description du mécanisme JWT

4.5.3 Cartographie avec Leaflet

Pour permettre de mettre les propriétés sur une carte, j'ai utilisé la bibliothèque JS Leaflet.

Leaflet est conçu dans un souci de simplicité , de performance et de convivialité . Il fonctionne efficacement sur toutes les principales platesformes de bureau et mobiles. [33]

4.5.4 Charte graphique

Le but de la charte graphique est de conserver une cohérence graphique dans les réalisations graphiques de l'entreprise. Ce qui permet aux agences d'identifier

Formes et lignes	Couleurs	Typographie
On a utilisé des formes rectangulaires et arrondies, simple instanciation des formes arrondies et circulaires du logo pour enrichir le graphisme.	<p>On a choisi comme couleurs :</p> <ul style="list-style-type: none"> — Le bleu ciel : Couleur faisant partie du logo et aussi porteuse d'optimisme et d'ouverture d'esprit. — Le gris light : plus proche du blanc que du noir afin d'assurer le calme et l'élégance — Celles des icônes utilisées : dans leur couleur de base, terme de simplicité et d'attraction. 	La typographie est « Tajawal » : une police moderne, nette, visible et simple qui garantit la clarté et une bonne lecture.

TABLE 4.8 – La charte graphique

facilement BnB PRO et, par habitude si la cohérence est respectée, de se repérer visuellement dans les différentes interfaces graphiques (ce qui facilite la navigation). Afin de respecter l'identité visuelle de BnB PRO, on a défini les formes et les lignes, la typographie utilisée, les codes couleurs et les visuels adoptés dans le tableau ci-dessous :

4.5.5 Editeur WYSIWYG Ckeditor

Pour assurer la création du modèle de contrat personnalisé, j'ai intégré cet éditeur WYSIWYG (voir la définition à la page 9).

En effet Ckeditor est conçu pour simplifier la création de contenu Web en offrant une interface utilisateur conviviale, intuitive et attractive.[34]

4.6 Conclusion

Dans ce chapitre, on a présenté l'environnement de travail (logiciel et matériel) et les langages qui ont permis de réaliser cette application.

Et enfin on a présenté quelques captures d'écran de l'application, une brève explication de leur fonctionnement et présenté quelques particularités de l'application.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Le digital prend de plus en plus de place dans le quotidien et le secteur de l’immobilier n’est pas épargné par cette transition. C’est pourquoi, afin de rester dynamique sur le marché, il est important pour les agences immobilière de faciliter leur activités grâce à un système complet et pratique. Avoir une application de ce genre assure une stratégie marketing complète et à la pointe des évolutions du marché.

Ce projet de fin d’étude passé au sein de la société McZEN TECHNOLOGIES avait pour but de réaliser une application web couvrant entièrement les besoins métiers fonctionnels et techniques et qui répond totalement aux exigences des utilisateurs finaux.

En effet, après avoir étudié et apporté des critiques au système actuel que possède ladite société, on a élaboré la conception de la solution et ensuite présenté les différents besoins fonctionnels et non fonctionnels de cette application pour enfin la réaliser tout en respectant une méthodologie Agile à savoir Kanban.

Concrètement, on avait une partie backend qui consistait à développer des Microservices relatives aux fonctionnalités CRUD de l’agence, les utilisateurs, les tâches, les clients, les contrats, etc... et une partie frontend où on a développer des interfaces offrant une bonne expérience utilisateur pour communiquer avec ces Microservices.

Notre application a réussi à apporter une gestion simple et efficace des activités des professionnels de l’immobilier et un gain de temps et d’énergie grâce à cette solution innovante. Elle leur permet aussi de se concentrer davantage sur le métier : consacrer plus de temps aux clients, la gestion des biens...

En perspectives, on pourra développer de nouvelles fonctionnalités comme la création de rôle et d’équipe, la récupération de mot de passe sans oublier une version mobile et ainsi apporter des modifications qui permettront à la société d’offrir plus de possibilité à ses clients et par la même occasion de gagner plus de marché dans le futur.

BIBLIOGRAPHIE ET NÉTOGRAPHIE

- [1] Bnb pro, 2021. <https://pro.bnbtunisie.com/>.
- [2] Bnb tunisie, 2021. <https://www.bnbtunisie.com/>.
- [3] Mczen technologies, 2021. <https://mc-zen.com/>.
- [4] Cto (chief technical officer - ou directeur des nouvelles technologies), 2021. <https://www.clementine.jobs/fiches-metiers/metiers-de-l-informatique/metier-chief-technical-officer-cto/>.
- [5] Testez, analysez et optimisez votre site web, 2021. <https://www.dareboost.com/fr/>.
- [6] Qu'est ce qu'un framework, 2021. <https://www.wildcodeschool.com/fr-FR/blog/framework-definition-developpement-web-programmation>.
- [7] Christian Soutou. *Apprendre SQL avec MySQL*. édition Eyrolles, 2006.
- [8] Besoins fonctionnels, 2015. https://www.memoireonline.com/11/12/6484/m_Conception-et-realisation-d'une-application-de-gestion-des-marches-par-appel-d'offres-aussein6.html.
- [9] Méthodes d'analyse et de conception, 2019. https://fr.wikipedia.org/wiki/M%C3%A9thode_d'analyse_et_de_conception.
- [10] Jira sofware, 2021. <https://www.atlassian.com/fr/software/jira>.
- [11] Qu'est-ce que la méthodologie kanban ?, 2021. <https://www.planzone.fr/blog/quest-ce-que-la-methodologie-kanban>.
- [12] Visualparadigme, 2021. <https://online.visual-paradigm.com/fr/>.
- [13] David Gabay Joseph Gabay. *UML 2 ANALYSE ET CONCEPTION*. éditions DUNOD, 2008.
- [14] Débutez l'analyse logicielle avec uml, 2019. <https://openclassrooms.com/fr/courses/2035826-debutez-lanalyse-logicielle-avec-uml/2035851-uml-c-est-quoi>.
- [15] Définition des diagrammes de cas d'utilisation uml 2.0, 2013. <http://doc-wiki.embarcadero.com/RADStudio/Rio/fr/D>

BIBLIOGRAPHIE ET NÉTOGRAPHIE

- [16] Uml : Les diagrammes de séquence, 2018.
<http://tvaira.free.fr/dev/uml/UML-DiagrammesDeSequence.pdf>.
- [17] Développement front-end et back-end : Quelles différences ?, 2021.
<https://blog.lesjeudis.com/developpement-front-end-et-back-end-quelles-differences>.
- [18] Environnement (informatique), 2019. [https://fr.wikipedia.org/wiki/Environnement_\(informatique\)](https://fr.wikipedia.org/wiki/Environnement_(informatique))
- [19] Spring tools 4, 2021. <https://spring.io/tools>.
- [20] Atom, 2021. <https://atom.io/>.
- [21] Texniccenter, 2021. <https://www.texniccenter.org/about/about-texniccenter>.
- [22] The collaboration platform for api development), 2021.
<https://www.postman.com/>.
- [23] Mysql workbench), 2021. <https://www.mysql.com/fr/products/workbench/>.
- [24] Thierry Templier. *Spring par la pratique*. éditions Eyrolles, 2009.
- [25] Hibernate, 2019. <https://www.jmdoudoux.fr/java/dej/chap-hibernate.htm>.
- [26] Spring boot, 2021. <https://spring.io/projects/spring-boot>.
- [27] Spring security, 2021. <https://spring.io/projects/spring-security>.
- [28] The progressive javascript framework, 2021. <https://vuejs.org/>.
- [29] Build fast, responsive sites with bootstrap, 2021. <https://getbootstrap.com/>.
- [30] Staruml, a sophisticated software modeler for agile and concise modeling, 2021. <https://staruml.io/>.
- [31] Plateforme gitlab devops, 2021. <https://about.gitlab.com/fr-fr/>.
- [32] Spring data jpa, 2021. <https://spring.io/projects/spring-data-jpa>.
- [33] Leaflet, an open-source javascript library for mobile-friendly interactive maps, 2021. <https://leafletjs.com/>.
- [34] Wysiwyg editor of tomorrow, available today, 2021. <https://ckeditor.com/>.

Annexes

Annexe A : Les pratiques de Kanban

La visualisation : afin de comprendre comment fonctionnent les dispositifs en place et connaître l'état du projet, il est essentiel de visualiser le flux de travail (workflow). Pour cela, utilisez un tableau dont chaque colonne représente une étape (à faire, ouvert, en cours, en test, terminé). Chaque tâche évolue jusqu'à ce qu'elle soit achevée.

La limitation du nombre de tâches en cours : chaque étape du tableau ne peut contenir qu'un nombre maximum de tâches en même temps, défini en fonction des capacités de l'équipe. Lorsqu'une tâche est terminée, une nouvelle peut alors être ajoutée.

La gestion du flux : il est essentiel de suivre, mesurer et consigner le déroulement du travail à travers chaque étape du tableau. Le but est de connaître la vitesse et la fluidité du travail.

L'explication des normes de processus : les règles du système Kanban doivent être formulées clairement et sans ambiguïté afin de s'assurer que l'équipe comprenne le travail réalisé et les améliorations futures.

L'identification des opportunités d'amélioration : une fois que l'équipe a compris les théories sur le travail, les processus et les risques, elle sera capable de discuter d'un problème ou d'un blocage auquel elle est confrontée et de trouver des améliorations à mettre en place.

Annexe B : Création de projet et de fiches Kanban avec Jira

Voici le scénario pour la création de projet et des fiches Kanban sur Jira Software.

En premier la page pour la sélection du type de projet avec une description de chacun des choix possibles.

Sélectionner le type de projet

Pourquoi existe-t-il deux types de projet ?

Les projets gérés par l'équipe représentent l'expérience Jira Software Cloud repensée de A à Z. Remarque : le choix que vous faites ici est définitif. La seule façon de changer de type de projet consiste à en créer un.

The screenshot displays the 'Select project type' interface. It features two main sections: 'Projet géré par l'entreprise' (left, dark blue background) and 'Projet géré par l'équipe' (right, blue background). Each section includes a brief description, a small icon, and a 'Selectionner l'expérience gérée par [type]' button at the bottom.

Type de Projet	Description	Icone	Action
Projet géré par l'entreprise	Les projets gérés par l'entreprise représentent ce qui a fait la renommée de Jira Software et sont là pour permettre aux fans de Jira de longue date de conserver leurs fonctionnalités préférées.	Icon showing a grid and various project management icons.	Sélectionner l'expérience gérée par l'entreprise
Projet géré par l'équipe	Grâce aux projets gérés par l'équipe, nous repensons Jira Software de A à Z afin de le rendre plus simple à configurer, à utiliser et à administrer pour les équipes de développement logiciel de toute taille.	Icon showing a graph and a progress bar.	Sélectionner l'expérience gérée par l'équipe

FIGURE 8 – Sélection et description des type de projet Jira

BIBLIOGRAPHIE ET NÉTOGRAPHIE

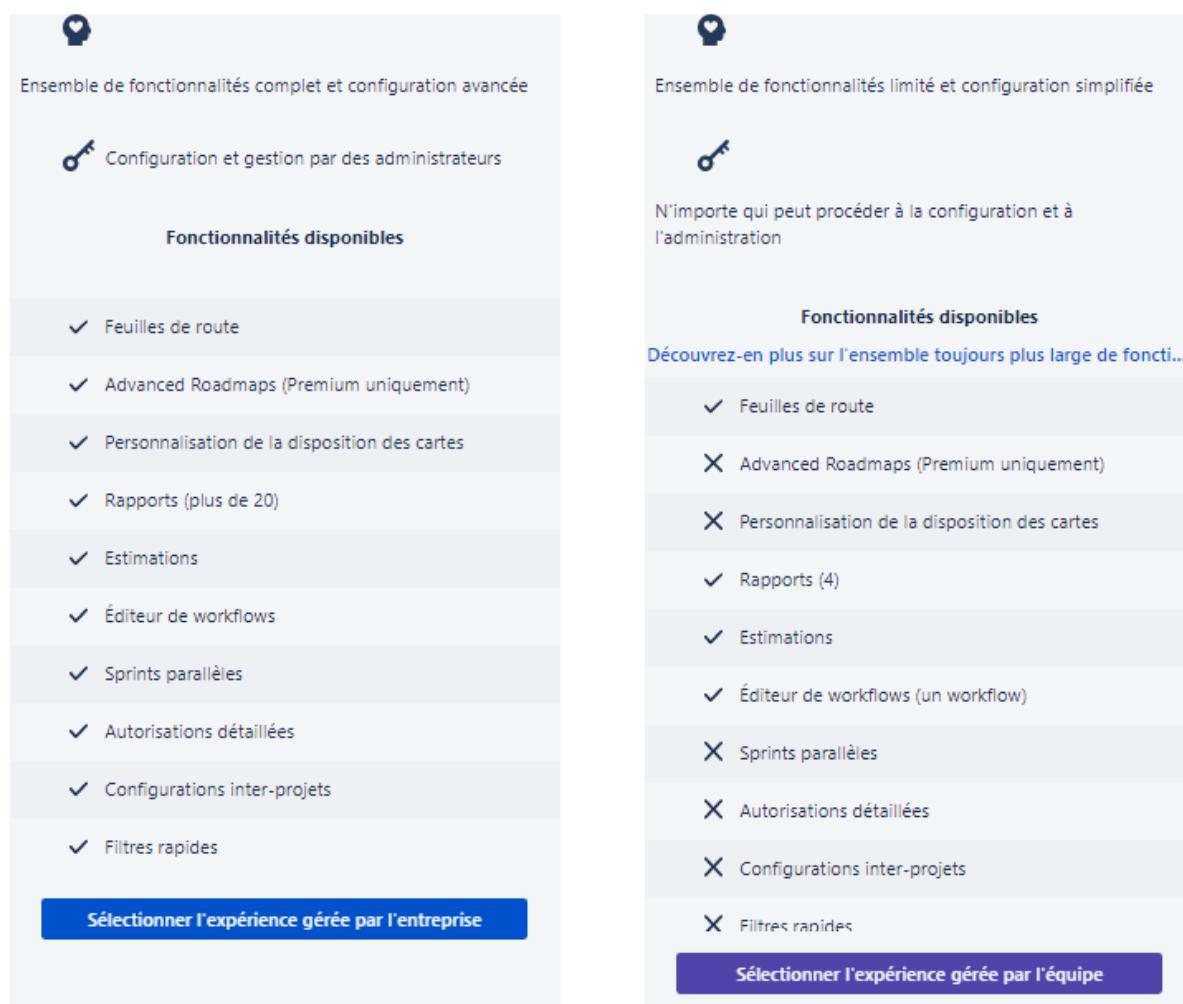


FIGURE 9 – Sélection et description des type de projet Jira

Après avoir sélectionné le type de projet ("Projet géré par l'entreprise" pour ce cas), on choisit la méthodologie Agile adoptée (Kanban).

Choisissez un modèle géré par l'entreprise

Les modèles gérés par l'entreprise offrent les fonctionnalités puissantes et pratiques dont vous avez besoin. Ils sont créés et gérés de façon centrale par un administrateur Jira grâce à des systèmes.

Développement ▾ Importer votre travail



Kanban
Surveillez le travail de manière continue pour les équipes Agile. Convient aux équipes contrôlant leur volume de travail depuis un backlog.

De quoi s'agit-il ? Sélectionner



Scrum
Gérez les stories, tâches et workflows d'une équipe Scrum. Pour les équipes livrant régulièrement des travaux.

De quoi s'agit-il ? Sélectionner



Suivi de bugs (auparavant développement logiciel de base)
Gérez la liste des tâches de développement et des bugs. Parfait pour les équipes qui n'ont pas besoin d'un tableau.

De quoi s'agit-il ? Sélectionner

Vous souhaitez une configuration facile et des fonctionnalités repensées ? [Essayer un modèle géré par l'équipe](#)

FIGURE 10 – Choix de la méthodologie Agile

On passe ensuite à la définition du nom du projet et sa création.

Créer un projet

Nom

BnB PRO

Clé 

BP

 Partager les paramètres avec un projet existant

Modèle



Kanban

Surveillez le travail de manière continue pour les équipes Agile. Convient aux équipes contrôlant leur volume de travail depuis un backlog.

[Modifier le modèle](#) Connectez le travail dans tous vos outils ▾

Synchronisez le travail à partir d'autres outils afin que votre équipe puisse accéder facilement au code, aux déploiements, au wiki, et plus encore depuis votre projet Jira.

[Créer](#)[Fermer](#)

FIGURE 11 – Nom et création du projet

Et enfin sur le tableau de bord, on peut passer à création des fiches Kanban appelées tickets et à leur suivi.

The screenshot shows a Jira Software dashboard titled "Tableau BNBPRO". The interface includes a header with navigation links like "Votre travail", "Projets", "Filtres", "Tableaux de bord", "Personnes", "Appli", and a "Créer" button. A search bar and user icons are also present.

The main content area displays five columns of issues:

- TACHE 1 ISSUE:** Gestion des paiements (BNBPRO-8)
- A FAIRE 1 ISSUE:** Gestion des demandes de propriétés (BNBPRO-4)
- EN COURS 2 ISSUES:**
 - Gestion des paramétrages (Matériel, TypeOffre, Équipement, TypePropriété, ModèleContrat) (BNBPRO-9)
 - Gestion des dépenses (BNBPRO-6)
- A TESTER 2 ISSUES:**
 - Gestion des contrats (BNBPRO-5)
 - Gestion des tâches (BNBPRO-7)
- FINI 3 ISSUES:**
 - Gestion des utilisateurs (Signup, AddCollab, Signin, Authentification) (BNBPRO-1)
 - Gestion des clients (BNBPRO-2)
 - Gestion de la propriétés (BNBPRO-3)

FIGURE 12 – Tableau de bord Jira

Annexe C : Référencement du site au niveau de différents moteurs de recherche.

Capture d'écran du résultat de recherche de BnB PRO sur Google.

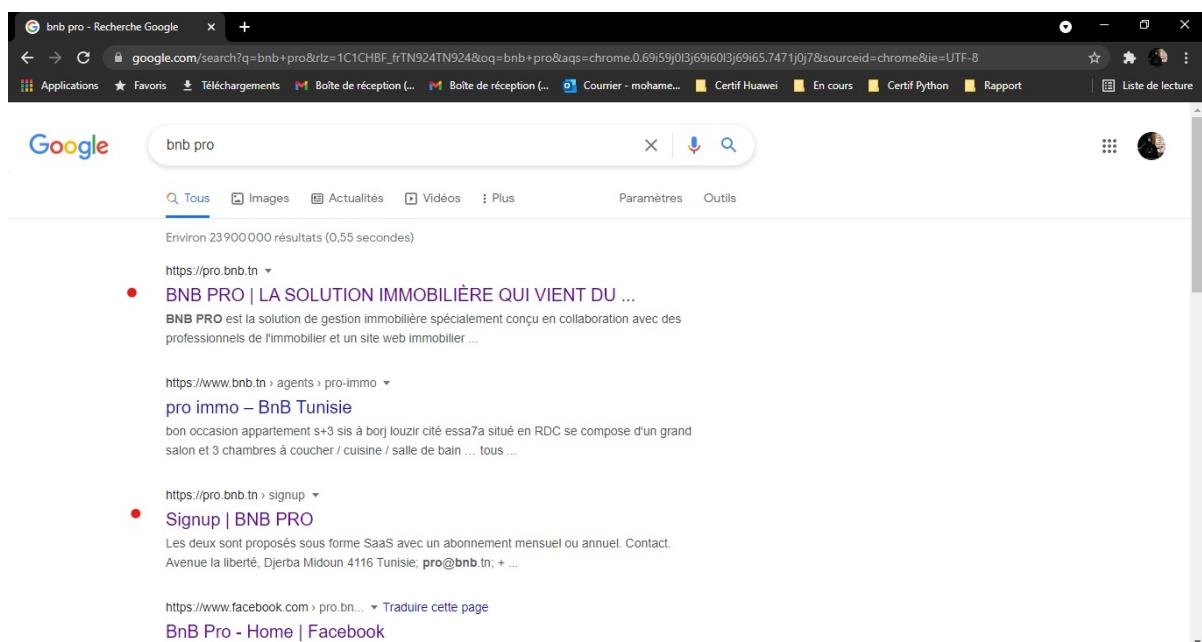


FIGURE 13 – BnB PRO sur Google

BIBLIOGRAPHIE ET NÉTOGRAPHIE

Capture d'écran du résultat de recherche de BnB pro sur Yahoo !

Annonce relative à: bnb pro

B&Bs in Are - Lowest Price Guarantee
www.booking.com
No reservation costs. Great rates. No Reservation Costs. Great Rates

No Booking Fees
Half-Price Hotels
Book online, Pay at Hotel.

Book Now
Quick, Simple, Easy to Use.
No reservation costs. Great rates.

Secure Booking
Easy and Secure Online Booking.
Read Real Reviews and Book Now!

Book for Tomorrow
Easy, fast and secure booking!
New deals listed every day

Book for Tonight
Your booking instantly confirmed!
Around-the-clock customer service

No Booking Fees
Half-Price Hotels
Book online, Pay at Hotel.

● **BNB PRO | LA SOLUTION IMMOBILIÈRE QUI VIENT DU FUTURE**
pro.bnbtv
BNB PRO est la solution de gestion immobilière spécialement conçue en collaboration avec des professionnels de l'immobilier et un site web immobilier personnalisable et optimisé pour le

FIGURE 14 – BnB PRO sur Yahoo !

Capture d'écran du résultat de recherche de BnB pro sur Bing.

Résultats pour **bnb pro** inclus.

Voulez-vous voir les résultats uniquement pour **bnb pro** ?

BNP Paribas Professionnels | Banque et assurance
<https://mabanquepro.bnpparibas>

Devenez client professionnel BNP Paribas. Vous démarrez une activité, nous vous accompagnons. Chez BNP Paribas, vous pouvez ouvrir et gérer un compte **professionnel** en ligne pour faciliter votre gestion au quotidien. Remplissez le formulaire et découvrez nos solutions associées à l'ouverture d'un compte **pro**.

Déconnexion
Contact. Nos conseillers vous répondent par téléphone, chat, mail ou bien encore grâce à ...

Le Groupe BNP Paribas
Toutes les informations sur BNP Paribas, banque internationale : offres d'emploi, ...

Projet De Crédit
Contact. Nos conseillers vous répondent par téléphone, chat, mail ou bien encore grâce à ...

Crédit De Campagne
X fermer éligibilité Pour souscrire un Crédit de Campagne, vous devez être (2) client BNP ...

Crédit Pro En Ligne
Vous pouvez emprunter sur une durée de 24 mois à 60 mois dans le cadre de votre Crédit ...

Financements De Trésorerie
COVID - 19. BNP Paribas propose le Prêt Garanti par l'Etat à ses clients ...

Afficher plus ▾

● **BNB PRO | LA SOLUTION IMMOBILIÈRE QUI VIENT DU FUTURE**
<https://pro.bnbtv>

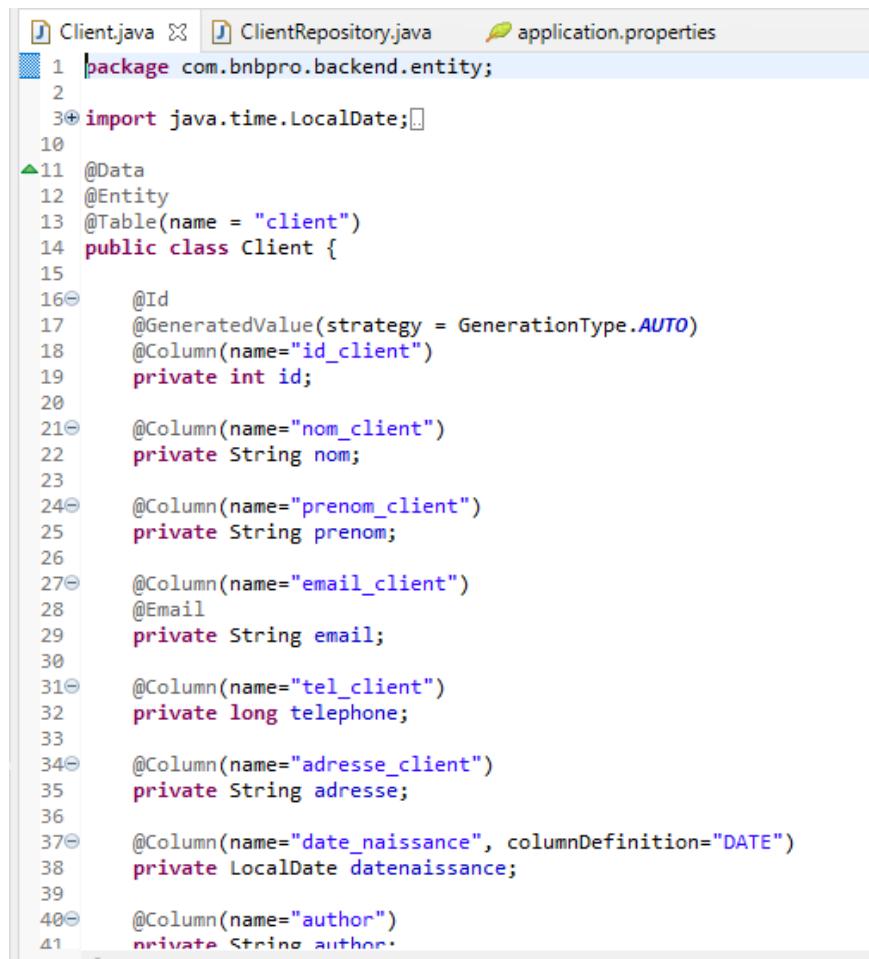
BNB PRO | LA SOLUTION IMMOBILIÈRE QUI VIENT DU FUTURE. Gestion Immobilière. Une solution 100% cloud, pour vous permettre de gérer vos propriétés, contrats, paiements reçus, dépenses et surtout...

FIGURE 15 – BnB PRO sur Bing

Annexe D : Description du fonctionnement de la persistance de données au niveau de Spring.

Pour la démonstration, on a choisi la classe JavaBean "Client" et son interface Repository pour montrer l'exemple.

Capture d'écran de la classe "Client".



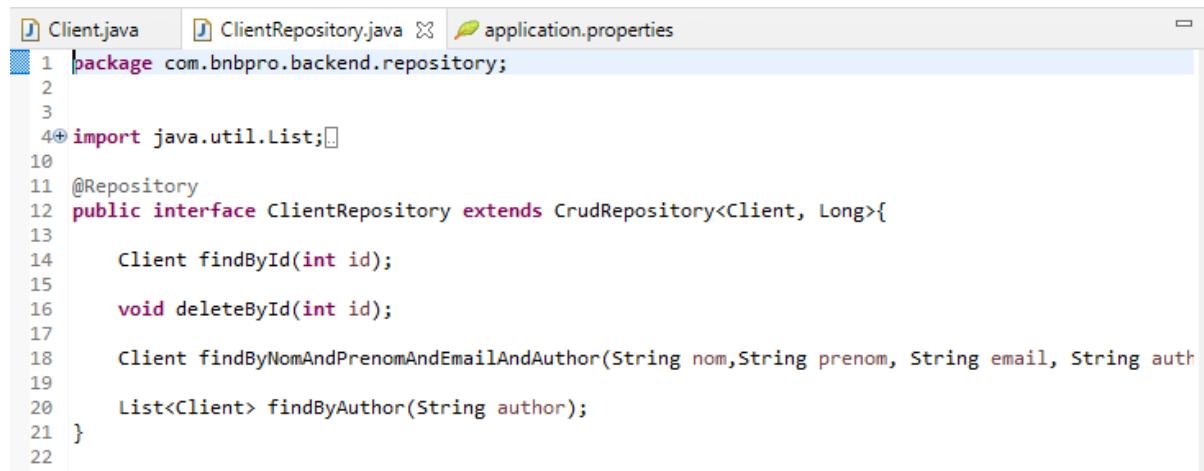
The screenshot shows a code editor with the following tabs at the top: Client.java, ClientRepository.java, and application.properties. The Client.java file is open and displays the following Java code:

```
1 package com.bnbprom.backend.entity;
2
3 import java.time.LocalDate;
4
5 @Data
6 @Entity
7 @Table(name = "client")
8 public class Client {
9
10    @Id
11    @GeneratedValue(strategy = GenerationType.AUTO)
12    @Column(name="id_client")
13    private int id;
14
15    @Column(name="nom_client")
16    private String nom;
17
18    @Column(name="prenom_client")
19    private String prenom;
20
21    @Column(name="email_client")
22    @Email
23    private String email;
24
25    @Column(name="tel_client")
26    private long telephone;
27
28    @Column(name="adresse_client")
29    private String adresse;
30
31    @Column(name="date_naissance", columnDefinition="DATE")
32    private LocalDate datenaissance;
33
34    @Column(name="author")
35    private String author;
```

FIGURE 16 – Classe "Client"

BIBLIOGRAPHIE ET NÉTOGRAPHIE

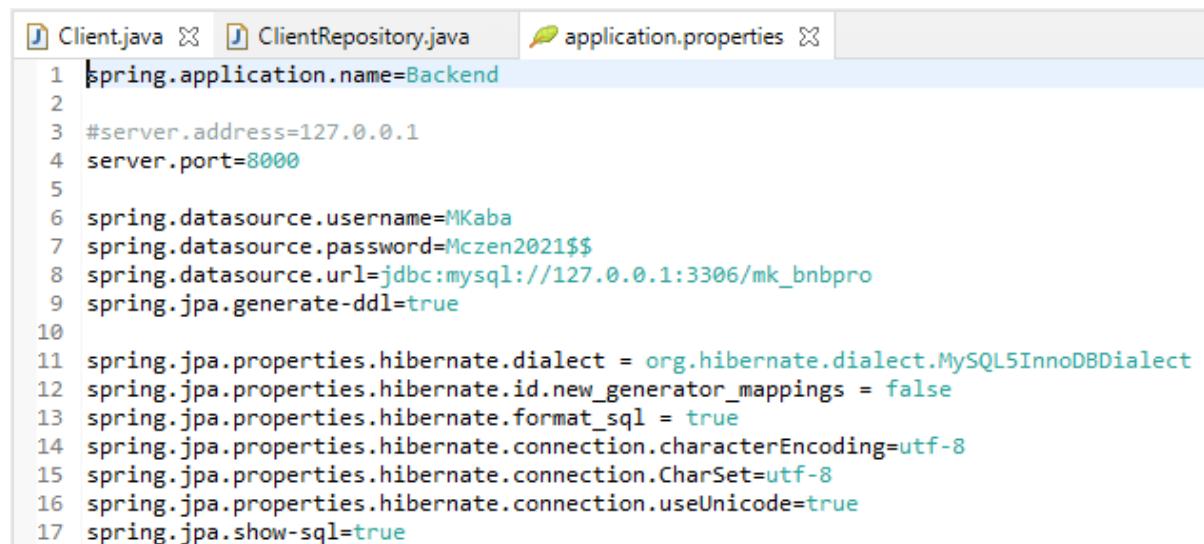
Capture d'écran de l'interface "ClientRepository".



```
Client.java ClientRepository.java application.properties
1 package com.bnipro.backend.repository;
2
3
4 import java.util.List;
5
6 @Repository
7 public interface ClientRepository extends CrudRepository<Client, Long>{
8
9     Client findById(int id);
10
11     void deleteById(int id);
12
13     Client findByNomAndPrenomAndEmailAndAuthor(String nom, String prenom, String email, String auth);
14
15     List<Client> findByAuthor(String author);
16
17 }
18
19
20 }
```

FIGURE 17 – Interface "ClientRepository"

Et la capture d'écran du fichier de configuration "application.properties".



```
Client.java ClientRepository.java application.properties
1 spring.application.name=Backend
2
3 #server.address=127.0.0.1
4 server.port=8000
5
6 spring.datasource.username=MKaba
7 spring.datasource.password=Mczen2021$$
8 spring.datasource.url=jdbc:mysql://127.0.0.1:3306/mk_bnipro
9 spring.jpa.generate-ddl=true
10
11 spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect
12 spring.jpa.properties.hibernate.id.new_generator_mappings = false
13 spring.jpa.properties.hibernate.format_sql = true
14 spring.jpa.properties.hibernate.connection.characterEncoding=utf-8
15 spring.jpa.properties.hibernate.connection.CharSet=utf-8
16 spring.jpa.properties.hibernate.connection.useUnicode=true
17 spring.jpa.show-sql=true
```

FIGURE 18 – Fichier de configuration "application.properties"

Annexe E : Communication avec les services via Postman.

Création d'un nouveau compte utilisateur et agence via la methode POST.

The screenshot shows the Postman interface for a POST request to `localhost:8000/signup`. The request body is JSON-formatted, containing fields for agence name, first name, last name, email, password, and telephone. The response status is 200 OK, with a timestamp of 321 ms and a size of 869 B. The response body is also JSON, showing the created user and agence details.

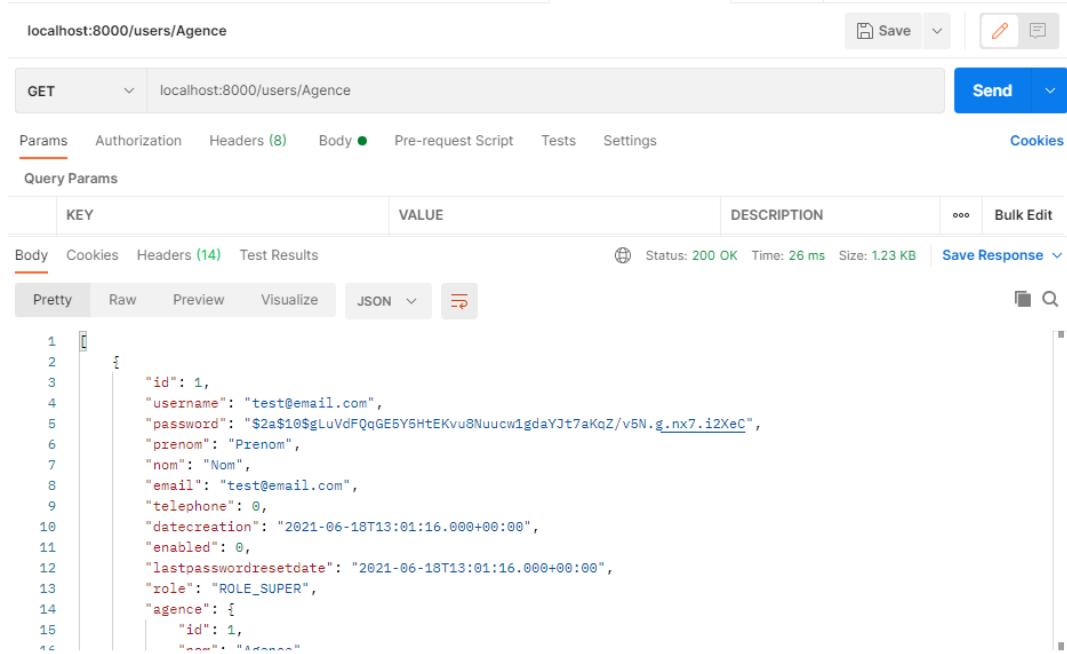
```
2   "nom_agence": "testnom_agence",
3   "firstname": "testprenom",
4   "lastname": "testnom",
5   "email": "test@test.com",
6   "password": "testpassword",
7   "telephone": ""
```

```
4   "password": "$2a$10$8jEzbCDDlidxNDGwRKw3c0ApB9og236ix6Hz2.17yqPLyBG4ylujG",
5   "prenom": "testprenom",
6   "nom": "testnom",
7   "email": "test@test.com",
8   "telephone": 0,
9   "datecreation": "2021-06-21T03:29:12.642+00:00",
10  "enabled": 0,
11  "lastpasswordresetdate": "2021-06-21T03:29:12.642+00:00",
12  "role": "ROLE_SUPER",
13  "agence": {
```

FIGURE 19 – La methode POST

BIBLIOGRAPHIE ET NÉTOGRAPHIE

Récupération de la liste des utilisateurs d'une agence via la méthode GET.

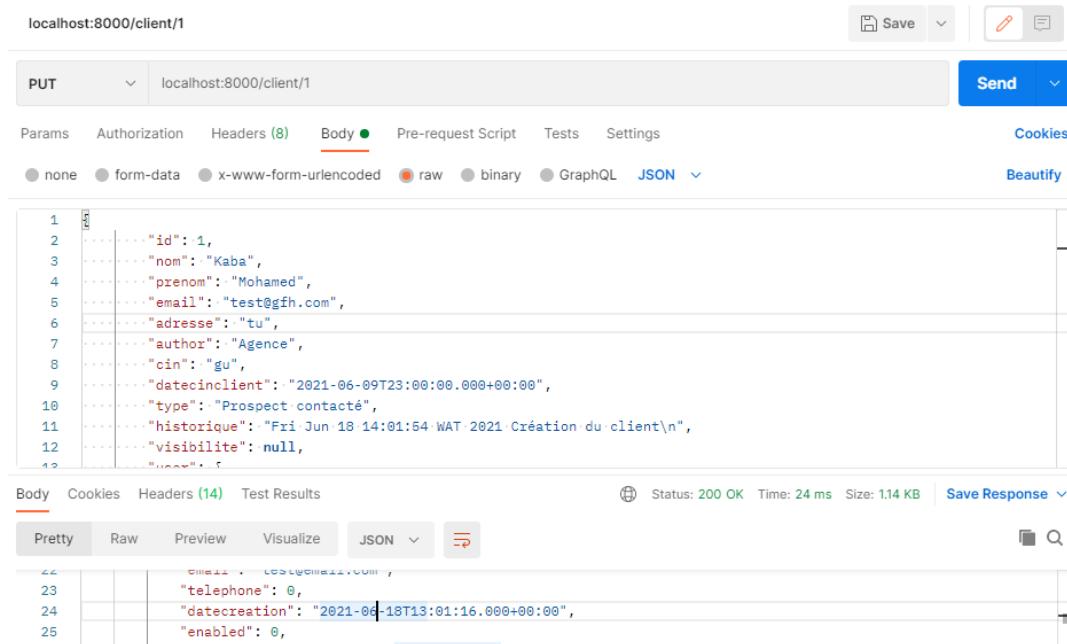


The screenshot shows a POSTMAN interface with the following details:

- URL: `localhost:8000/users/Agence`
- Method: `GET`
- Headers: `Authorization`, `Headers (8)`, `Body` (selected), `Pre-request Script`, `Tests`, `Settings`, `Cookies`
- Query Params: None
- Body: `Pretty` view shows a JSON object with fields: `id`, `username`, `password`, `prenom`, `nom`, `email`, `telephone`, `datecreation`, `enabled`, `lastpasswordresetdate`, `role`, `agence`. The `agence` field contains a nested object with `id` and `nom`.
- Response: Status: `200 OK`, Time: `26 ms`, Size: `1.23 KB`, `Save Response` button

FIGURE 20 – La méthode GET

Modification des informations d'un client via la méthode PUT.



The screenshot shows a POSTMAN interface with the following details:

- URL: `localhost:8000/client/1`
- Method: `PUT`
- Headers: `Authorization`, `Headers (8)`, `Body` (selected), `Pre-request Script`, `Tests`, `Settings`, `Cookies`
- Body: `none` (radio button selected), `form-data`, `x-www-form-urlencoded`, `raw` (radio button selected), `binary`, `GraphQL`, `JSON` (selected), `Beautify` button
- Body: `Pretty` view shows a JSON object with fields: `id`, `nom`, `prenom`, `email`, `adresse`, `author`, `cin`, `dateclient`, `type`, `historique`, `visibleite`. The `author` field is set to "Agence".
- Response: Status: `200 OK`, Time: `24 ms`, Size: `1.14 KB`, `Save Response` button

FIGURE 21 – La méthode PUT

BIBLIOGRAPHIE ET NÉTOGRAPHIE

Suppression d'un modèle de contrat via la méthode DELETE.

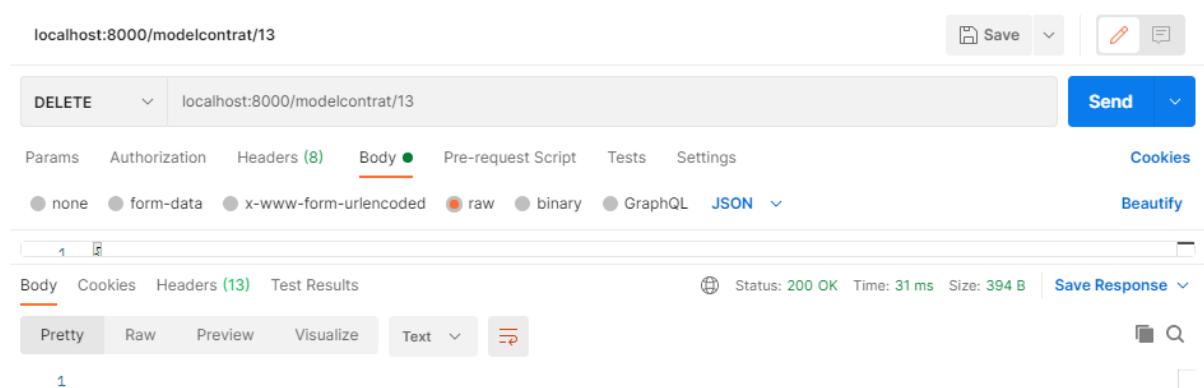


FIGURE 22 – La méthode DELETE