

Province de Hainaut
IETC - Mercure
Enseignement Supérieur
Economique de Promotion Sociale

Bachelier en informatique de gestion

Epreuve intégrée

MOBEMBO.CD

Application de e-commerce
de mise en location de biens immobiliers

Tshibangu Cédric
Année scolaire : 2021 – 2022

Chargé de cours de l'épreuve intégrée :
Alain MBAYO BOGA ASANI KALUNGA

Remerciements

Après avoir passé plus ou moins 10 ans à travailler dans le secteur alimentaire, il m'a semblé plus qu'évident d'emprunter une autre voie, de changer mon statut professionnel et de ce fait, améliorer ma situation familiale et relever mon statut social.

C'est pour cette raison qu'en juin 2018, j'ai pris la décision d'entamer un bachelier en cours du soir. Je me suis lancé dans cette aventure de 3 ans pour obtenir une plus-value personnelle.

Ce fût une aventure semée d'embûches, tant sur le plan physique que psychologique. Lorsque j'ai pris la décision d'entamer un bachelier en informatique de gestion, j'étais loin d'imaginer toutes les concessions et sacrifices qui m'attendaient.

C'est pourquoi je dédie cette réussite à mon épouse. Elle qui a été ma boussole dans cette tempête et sur qui je pouvais m'appuyer quand il m'arrivait de mettre genoux au sol. Je remercie également mes enfants pour leur bienveillance durant ces longues années. Nous avons beaucoup de temps à rattraper...

Une pensée sincère à la direction de l'Enseignement Supérieur Economique de Promotion Sociale qui a su tout mettre en œuvre pour me fournir un environnement sain, même pendant la pandémie.

Sans oublier les professeurs rencontrés, que vous soyez de l'IETC ou des Aumôniers du Travail, je vous remercie du fond du cœur pour l'énergie dispensée afin de me transmettre votre savoir. J'adresse également mes remerciements à toutes les personnes du centre de formation Technofutur Tic de Gosselies. Vous qui m'avez apporté plus qu'un savoir-faire. Grâce à vous, j'ai pu acquérir une méthodologie indispensable dans un environnement orienté objet, ainsi qu'une grande compréhension des langages tels que Java, Angular et .NET.

Et toutes ces personnes rencontrées durant ces quatre années, celles et ceux qui ont su trouver les mots justes pour me motiver à aller toujours plus loin, sans vous je ne serais peut-être pas arrivé à mon niveau de connaissance actuel

Merci à toutes et à tous...

 Signature récupérable

X 

Tshibangu Cédric
Etudiant en Informatique de Gestion
Signé par : 1ce5dcbc-7166-41ab-85c4-574be1b1358b

Table des matières

1. Introduction	8
1.1. Mobembo.cd	8
1.2. Explication de la problématique	8
1.3. Les constats	9
1.4. Le saviez-vous ?	9
1.5. Proposition de solution	11
2. Abstract.....	12
3. Objectif	12
4. Le cahier des charges.....	13
4.1 Introduction	13
4.2 Présentation de l'entreprise.....	13
4.3 Objectif de l'entreprise.....	13
4.4 La population ciblée	13
4.5 Fonctionnalités attendues.....	14
4.6 Environnement du système.....	14
4.7 Les limites du système.....	15
4.8 Conclusion	15
5 Analyse.....	16
5.1 Introduction	16
5.2 Diagramme de cas d'utilisation	16
5.3 Diagramme de navigation - Consulter	17
5.3.1 Cas d'utilisation : Consulter	18
5.4 Diagramme de navigation - S'inscrire	19
5.4.1 Cas d'utilisation : S'inscrire.....	20
5.5 Diagramme de cas d'utilisation client.....	21
5.5.1 Diagramme de navigation client – Se connecter	22
5.5.1.1 Cas d'utilisation : Se connecter	22
5.5.2 Diagramme de navigation client - Modification du mot de passe.....	23
5.5.2.1 Cas d'utilisation : Modification du mot de passe.....	23
5.5.3 Diagramme de navigation client - Gestion de compte	24
5.5.3.1 Cas d'utilisation : Enregistrement d'adresse	24
5.5.3.2 Cas d'utilisation : Enregistrement d'informations bancaires.....	25
5.5.4 Diagramme de navigation client - Gestion de Bien - Création	26
5.5.4.1 Cas d'utilisation gestion de bien : Création	26
5.5.5 Diagramme de navigation client - Gestion de Bien - Suppression.....	27
5.5.5.1 Cas d'utilisation gestion de bien : Suppression.....	27
5.5.6 Diagramme de navigation client - Gestion de Bien - Mis en ligne.....	29
5.5.6.1 Cas d'utilisation gestion de bien : Mis en ligne.....	29

5.5.7	Diagramme de navigation client - Gestion de Bien - Contrat mis en ligne	31
5.5.7.1	Cas d'utilisation gestion de bien : Contrat mis en ligne	31
5.5.8	Diagramme de navigation client - Gestion de Bien – Annulation contrat mis en ligne	32
5.5.8.1	Cas d'utilisation gestion de bien : Annulation contrat mis en ligne.....	32
5.5.9	Diagramme de navigation client - Gestion de Bien – Information sur le Bien.....	34
5.5.9.1	Cas d'utilisation gestion de bien : Information sur le bien	34
5.5.10	Diagramme de navigation client - Gestion de Bien – « Liker » un Bien.....	35
5.5.10.1	Cas d'utilisation gestion de bien : Like bien	35
5.5.11	Diagramme de navigation client - Gestion de réservation – Création	36
5.5.11.1	Cas d'utilisation gestion de réservation : Création	36
5.6	<i>Diagramme de cas d'utilisation admin</i>	38
5.6.1	Création du rôle administrateur	38
5.7	<i>Conclusion</i>	38
6	Développement :	39
6.1	<i>Introduction</i>	39
6.2	<i>Environnement de travail</i>	39
6.2.1	IntelliJ.....	39
6.2.2	WampServer	40
6.3	<i>Langage utilisé</i>	41
6.3.1	SQL.....	41
6.3.2	Java	41
6.3.3	Type script	41
6.4	<i>Installation et mise en route en local.</i>	42
6.4.1	Base de données.....	42
6.4.2	Rest-API Java.....	42
6.4.3	Client Web	42
6.5	<i>Le Backend</i>	43
6.5.1	Base de données.....	43
6.5.1.1	Base de données : introduction	44
6.5.1.2	Base de données : Événement.....	44
6.5.1.3	Base de données : Liste des tables	44
6.5.1.4	Base de données : Relations entre tables	45
6.5.1.4.1	Personne – Doc_Officiel	45
6.5.1.4.2	Personne – RolePersonne.....	45
6.5.1.4.3	Personne – Adresse_Personne	45
6.5.1.4.4	Personne – Contact_personne	46
6.5.1.4.5	Personne – PassWord.....	46
6.5.1.4.6	Personne – Info_Bancaire_personne	46
6.5.1.4.7	Personne – Contrat_mis_en_ligne	47
6.5.1.4.8	Personne – Contrat_Location	47
6.5.1.4.9	Personne – Bien.....	47
6.5.1.4.10	Personne – Bien (Like)	48
6.5.1.4.11	Bien – Image_bien	48
6.5.1.4.12	Bien – Type_Bien	48
6.5.1.4.13	Bien – Aladisposition	49
6.5.1.4.14	Bien - Coordonnee.....	49
6.5.1.4.15	Coordonnee – Ville	49
6.5.1.4.16	Service – Coordonnee.....	50
6.5.1.4.17	Service – Type_Service	50

6.5.1.4.18	Ville – Image_Ville	50
6.5.1.4.19	Ville – Province	51
6.5.1.4.20	Province – Image_Province	51
6.5.1.4.21	Adresse_Personne – Pays	51
6.5.2	REST-API	52
6.5.2.1	Diagramme de navigation des données	52
6.5.2.2	Maven	53
6.5.2.3	Application Properties	53
6.5.2.4	Architecture	54
6.5.2.4.1	Dossier Data_Access	55
6.5.2.4.1.1	Entity	55
@Entity	55
@Id	55
@GeneratedValue	55
@Column	55
@OneToMany	56
@OneToOne	56
@ManyToOne	56
@ManyToMany	56
@Getter	56
@Setter	56
@NoArgsConstructor	56
@AllArgsConstructor	56
6.5.2.4.1.2	Repository	57
6.5.2.4.2	Dossier Business	58
6.5.2.4.2.1	Dossier DTO	58
6.5.2.4.2.1.1	@ToString	58
6.5.2.4.2.1.2	IdentifiedDTO	58
6.5.2.4.2.2	Dossier Mapper	59
6.5.2.4.2.3	Dossier Service	59
6.5.2.4.3	Le dossier exception	61
6.5.2.4.3.1	La classe ElementAlreadyExistsException :	61
6.5.2.4.3.2	La classe ElementFoundException :	62
6.5.2.5	Les dossiers Rest-API	63
6.6	Client Web	64
6.6.1	Angular	64
6.6.1.1	Les librairies	64
6.6.2	Structure du code	64
6.6.2.1	Les Classes	64
6.6.2.2	Structure des « Components »	65
6.6.2.2.1	Les branches	66
6.7	Conclusion	67
7	Cybersécurité	68
7.1	Introduction	68
7.2	Définition	68
7.3	Protéger le système	70
7.3.1	Avec POO	70
7.3.2	Contre les utilisateurs	70
7.3.3	Contrôler les utilisateurs et protéger leurs données	71
7.3.3.1	JWT	71

7.3.3.1.1	La composition d'un JWT.....	72
7.4	<i>Conclusion</i>	72
8	Conclusion finale	73
	Bibliographie	74

Liste des figures

Figure 1 Où le tourisme est une moteur de l'économie.....	8
Figure 2 Cote Congolaise	9
Figure 3 Montagne Congolaise.....	9
Figure 4 Forêt tropicale Congolaise.....	9
Figure 5 plat Congolais : Pongu	10
Figure 6 plat Congolais : Chikwangue.....	10
Figure 7 plat Congolais : Fumbwa.....	10
Figure 8 plat Congolais : Mikaté	10
Figure 9 plat Congolais : Liboké	10
Figure 10 Commentaire : expert Rd Congo	10
Figure 11 Vue générale SYSTEME : Angular - Java - MySQL	12
Figure 12 Schéma Visio : Diagramme de cas d'utilisation	16
Figure 13 Schéma Visio : Diagramme de navigation - Consulter.....	17
Figure 14 Schéma Visio : Diagramme de navigation - S'inscrire.....	19
Figure 15 Schéma Visio : Diagramme de cas d'utilisation client	21
Figure 16 Schéma Visio : Diagramme de navigation client - Se connecter.....	22
Figure 17 Schéma Visio : Diagramme de navigation client - Modification mot-de-passe	23
Figure 18 Schéma Visio : Diagramme de navigation client - GESTION de compte	24
Figure 19 Schéma Visio : Diagramme de navigation client - Gestion de bien - création	26
Figure 20 Schéma Visio : Diagramme de navigation client - Gestion de bien - Suppression.....	27
Figure 21 Schéma Visio : Diagramme de navigation client - Gestion de bien - Mis en ligne.....	29
Figure 22 Schéma Visio : Diagramme de navigation client - Gestion de bien - Contrat mis en ligne.....	31
Figure 23 Schéma Visio : Diagramme de navigation client - Gestion de bien - Annulation contrat mis en ligne.....	32
Figure 24 SCHEMA VISIO : DIAGRAMME DE NAVIGATION CLIENT - GESTION DE BIEN - Information sur le bien.....	34
Figure 25 SCHEMA VISIO : DIAGRAMME DE NAVIGATION CLIENT - GESTION DE BIEN - Like bien	35
Figure 26 SCHEMA VISIO : DIAGRAMME DE NAVIGATION CLIENT - GESTION DE Réservation - création.....	36
Figure 27 Diagramme de cas d'utilisation admin	38
Figure 28 Image issu de google - IntelliJ IDEA	39
Figure 29 Image issu de google - WampServer	40
Figure 30 Image issu de google - MySQL.....	41
Figure 31 Image issu de google - JAVA / Spring.....	41
Figure 32 Image issu de google - Angular.....	41
Figure 33 Image issu de MySQL - Schéma base de données Mobembo.cd	43
Figure 34 Schéma Visio - Diagramme de navigation des données projet Spring Boot.....	52
Figure 35 Image issu de google - MAVEN / Spring	53
Figure 36 Schema VISIO - Plan backend	54
Figure 37 Image issu de google - Angular.....	64
Figure 38 Schéma Visio - Structure client Angular	65
Figure 39 Image issu de google - Cybersécurité	68
Figure 40 Image issu de google - Programmation Orienté Objet.....	70
Figure 41 Schéma Visio - Gestion des requêtes avec JWT.....	71

1. Introduction

1.1. Mobembo.cd

Étant originaire de la R.D.C, j'ai eu l'envie d'offrir une plus-value aux habitants du pays dont une des langues bantoues officielle est le Lingala. Nommer mon application " Mobembo " c'est-à-dire voyage et " CD" identification Web de la R.D.C, me semblait donc naturel.

1.2. Explication de la problématique



FIGURE 1 OÙ LE TOURISME EST UNE MOTEUR DE L'ECONOMIE

En 2019, le site *Statista* a publié l'article nommé « où le tourisme est un moteur de l'économie ». Cet article démontre que pour l'économie d'une nation, le tourisme est très important.¹ (1)

En lisant tous ces articles, je me suis demandé « à quel niveau se situait la République Démocratique du Congo ? ».

Pour la période 2019, aucun article ne fait le lien entre le PIB de la RDC et son tourisme. Mais en 2015, selon un article sur le blog de Radio Okapi « le tourisme a contribué à hauteur de 0,7% au PIB de la RDC ».² (2)

Certes, la RDC est l'un des pays les plus pauvres, mais elle possède tous les atouts pour faire partie des destinations touristiques les plus prisées.

¹ Gaudiaut, Tristan. Où le tourisme est un moteur de l'économie. *statista.com*. [En ligne] 29 avril 2020. <https://fr.statista.com/infographie/21543/contribution-du-tourisme-dans-le-pib-des-pays/>.

² Bompengo, Jahn. RDC: La contribution du tourisme au PIB est faible. *radiookapi.net*. [En ligne] 27 Septembre 2016. <https://www.radiookapi.net/2016/09/27/actualite/economie/rdc-la-contribution-du-tourisme-au-pib-est-faible#:~:text=Le%20tourisme%20a%20contribu%C3%A9%20%C3%A0,%2Dt%2Dil%20fait%20savoir..>

1.3. Les constats

Constat 1 : En 2019, le salaire minimum national en République Démocratique du Congo était de 7075 FFC (franc Congolais)³ (3) ce qui revient à 3,21€ par jour.

Constat 2 : 852€ est le budget que les Français vont consacrer en moyenne pour leurs vacances en 2019.⁴ (4)

Si nous nous fions à ces deux constats, nous pouvons aisément dire que les touristes occidentaux peuvent avoir une incidence réelle sur l'économie locale, voir nationale.

Les attentes d'un touriste :

Le dépaysement	Un lieu différent de son quotidien	La montagne	Découvrir de nouvelles choses
La plage	Les activités	La mer	Découvrir une autre culture

Mais la République Démocratique du Congo possède la plupart de ces atouts !

Effectivement, la République Démocratique du Congo est un monde à part, avec une culture bien ancrée.

On pense différemment, on agit différemment.

1.4. Le saviez-vous ?



FIGURE 2 COTE CONGOLAISE

La République Démocratique du Congo possède une côte de 170 km de long, qui donne directement accès à l'océan Atlantique ;



FIGURE 3 MONTAGNE CONGOLAISE

Le mont Stanley est la troisième montagne la plus haute d'Afrique avec un sommet à plus de 5 100 m d'altitude ;



FIGURE 4 FORET TROPICALE CONGOLAISE

La forêt tropicale du bassin du Congo s'étend sur plus de 3,6 millions de kilomètres carrés et six pays, du Cameroun à l'ouest à la République Démocratique du Congo en Afrique centrale.

³ Salaire minimum - Congo_kinshasa. [votresalaire.org](https://votresalaire.org/congo/salaire/salaire-minimum/archive/20190701). [En ligne] 1 juillet 2019. <https://votresalaire.org/congo/salaire/salaire-minimum/archive/20190701>.

⁴ Ouest-France. Vacances d'été. Avec 852 €, le budget moyen des Français en baisse. [ouest-france.fr](https://www.ouest-france.fr). [En ligne] 3 juillet 2019.

Sans oublier la gastronomie typique.

Pondu



FIGURE 5 PLAT CONGOLAIS :
PONDU

Chikwangue



FIGURE 6 PLAT
CONGOLAIS :
CHIKWANGUE

Fumbwa



FIGURE 7 PLAT
CONGOLAIS :
FUMBWA

Mikaté



FIGURE 8 PLAT
CONGOLAIS : MIKATE

Liboké




FIGURE 9 PLAT CONGOLAIS
: LIBOKE

Malheureusement, le Congolais est incapable à l'heure actuelle de penser ou de se préoccuper d'autre chose que de sa propre survie, tant les biens pour combler les besoins élémentaires viennent à manquer.

Les gens cherchent à se nourrir, à subsister. Ils ne font plus attention à la beauté qui les entoure et surtout au bénéfice que celle-ci pourrait leur rapporter. Comme preuve, je n'ai trouvé aucun Congolais pouvant me décrire sa province dans son entièreté : ses musées, ses coins touristiques ou autres, car chacun y va pour un intérêt personnel (famille, affaires...), mais malheureusement pas pour le pays.

Et pourtant :⁵ (5)



Notre expert Rd Congo
Caroline rédactrice alibabuy.com

Les Congolais sont très accueillants et il est toujours agréable de partager un moment avec eux autour par exemple d'une tasse de thé. Si vous allez à la rencontre d'ethnies vivant en dehors des grandes agglomérations, amenez-leur un petit présent tel que des noix de cola ou piments indigènes. Vous leur ferez ainsi plaisir et honneur.

FIGURE 10 COMMENTAIRE : EXPERT RD CONGO

⁵ Caroline. Guide touristique pour visiter la Rd Congo et bien préparer son voyage. *alibabuy.com*. [En ligne] https://www.alibabuy.com/guide-touristique/rd_congo.html.

1.5. Proposition de solution

Étant bientôt diplômé en informatique de gestion, je ne peux pas agir sur la politique du pays. Cependant, grâce aux compétences acquises durant ces quatre années, lors de mon bachelier, je pense que je peux peut-être agir d'une autre manière...

Comment ?

Quelle que soit la destination, un touriste se pose toujours la même question essentielle...

Où vais-je loger ?

C'est là que moi, développeur, je peux agir en réalisant une application qui permettrait de faciliter la recherche de logements en RDC. Pas seulement des entreprises hôtelières, mais surtout des avoirs de la population locale.

Mon but est de faire prendre conscience aux Congolais qu'ils possèdent tout ce qu'il faut pour attirer les touristes.

Mon application « *N'est qu'une goutte d'eau dans l'océan, mais sans gouttes d'eau il n'y aurait pas d'océan* ». Citation de Mère Térésa (1910-1997).

2. Abstract

Ce travail de fin d'étude apporte une solution aux problèmes que rencontre le secteur touristique en République Démocratique du Congo.

Cette solution a pour objectif la redynamisation du secteur touristique en RDC en permettant à ses habitants d'accueillir temporairement, et contre rémunération, des visiteurs au sein de leurs propriétés.

Cette solution est une application de e-commerce réalisée en Angular 13 ainsi qu'un Rest API en Java spring boot et d'une base de données en MySQL.



FIGURE 11 VUE GENERALE SYSTEME : ANGULAR - JAVA - MYSQL

3. Objectif

L'objectif du projet est également intimement lié à une volonté personnelle : apporter une aide pour décroître la pauvreté en République Démocratique du Congo.

Comment ?

En offrant au peuple congolais un service de location de biens (chambres, appartements, maisons, etc.)

Il en découle les avantages suivants :

1. Faire (re)découvrir des régions aux touristes.
2. Booster l'économie locale.

4. Le cahier des charges

4.1 Introduction

Pour le cahier des charges, je suis parti sur l'invention d'une ASBL spécialisée dans le domaine de l'import-export entre l'Europe et l'Afrique centrale. Cette ASBL désire participer au processus de développement du secteur du tourisme en Afrique centrale.

Dans ce chapitre j'aborde la présentation de l'entreprise ainsi que son objectif, la population ciblée, les fonctionnalités attendues, l'environnement du système et ses limites.

4.2 Présentation de l'entreprise

L'entreprise TSHISHI ASBL basée à Anvers, Belgique, aimerait varier ses activités en investissant dans le secteur de l'immobilier en Afrique centrale.

Comme premier objectif, l'ASBL a choisi de s'implanter en République Démocratique du Congo en tant que plateforme de location de bien.

Elle ne dispose pas de moyens suffisants pour la conception à la fois d'une application web et d'une application mobile, mais cherche à rallier les deux mondes pour limiter le coût de cette première expérience.

Pour la réalisation de ce projet, elle mettra à disposition un budget de 30 000€ qui couvrira la totalité des besoins pour la réalisation de l'application ainsi que son déploiement.

4.3 Objectif de l'entreprise

L'objectif de l'entreprise est le déploiement d'une application, la plus autonome possible.

Cette application mettra en relation tous les propriétaires possédant un bien disponible (maison, appartement, chambre, lit, mobile home, terrain...) avec toutes les personnes désirant louer.

4.4 La population ciblée

- Les entreprises de passage ;
- Les touristes (étrangers) désirant découvrir les coutumes locales ;
- La population locale désirant trouver rapidement un hébergement ;
- La tranche d'âge visée est de 18 ans à 75 ans.

4.5 Fonctionnalités attendues

L'entreprise THSISHI ASBL veut que toutes personnes se rendant sur l'application puissent avoir accès, par le biais d'un affichage simplifié, aux biens mis en ligne.

Ces personnes auront la possibilité de s'inscrire, ce qui leur permettra d'avoir plus d'options sur l'application. Par exemple, ils ont la possibilité de gérer leur compte (modification de mot de passe, enregistrement d'informations personnelles), de gérer leur(s) bien(s) (création, suppression, mis en ligne d'un bien, gestion de contrat liée à la mise en ligne, ...), de gérer les réservations (réservation de bien, gestion de contrat liée aux réservations, ...), ainsi que la possibilité de « liker » un bien ou de visualiser la fiche technique détaillée d'un bien mis en ligne.

L'application devra être capable de générer automatiquement un contrat lorsqu'un client (dans son rôle de propriétaire) met un bien en ligne ou lorsque le client (dans son rôle de loueur) fait une réservation. Elle devra aussi être capable de modifier un contrat si le client désire mettre fin à un contrat de mis en ligne ou de réservation. L'action de réservation de bien ou de résiliation de contrat entraînera des frais que le client devra régler sur l'application pour acter sa décision.

4.6 Environnement du système

L'application est conçue pour interagir avec deux types de clientèle :

1. Des personnes physiques
 - L'application peut être questionnée et peut répondre aux utilisateurs dans un cadre défini. Ces utilisateurs sont représentés par quatre profils : visiteur, futur client, client et administrateur (admin).
2. Des systèmes extérieurs de messagerie de type e-mail.
 - L'application est capable de communiquer, selon le processus d'une personne physique, des éléments de type e-mail à un service tiers.

4.7 Les limites du système

Fonctionnalités attendues	Disponible sur l'application
1. Affichage simplifié de bien mis en ligne	Oui
2. La possibilité de s'inscrire	Oui
3. La possibilité de modifier son mot de passe	Oui
4. La possibilité d'enregistrer son adresse personnelle	Oui
5. La possibilité d'enregistrer ses données bancaires	Oui
6. La possibilité de créer un bien	Oui
7. La possibilité de supprimer un bien	Oui
8. La possibilité de mettre un bien en ligne	Oui
9. La création d'un contrat lorsqu'un bien est mis en ligne	Oui
10. La possibilité de visionner le contrat d'un bien mis en ligne	Oui
11. La possibilité de résilier le contrat d'un bien mis en ligne	Oui
12. La possibilité de réserver un bien	Oui
13. La création d'un contrat lorsqu'un bien est réservé	Oui
14. La possibilité de visionner un contrat de réservation	Oui
15. La possibilité de résilier le contrat d'une réservation	Non
16. La possibilité de visionner la fiche technique détaillée d'un bien	Oui
17. La possibilité de « liker » un bien.	Oui

Les améliorations souhaitées
1. La possibilité de diffuser un lien sur les réseaux sociaux
2. La création d'une application Play store
3. La création d'une application Apple Store

4.8 Conclusion

Dans ce chapitre, j'aurais voulu avoir plus de temps pour la récolte d'informations concernant les lois dans le cadre de location de biens en République Démocratique du Congo. Ces informations auraient pu me servir pour adapter mon cahier des charges aux exigences du pays.

5 Analyse

5.1 Introduction

Dans cette analyse, je définis les acteurs et les interactions possibles avec le système, grâce à des diagrammes de cas d'utilisation, des diagrammes de navigation ainsi que la description de chaque scénario. Je délimite le cadre du système en décrivant chaque action possible.

5.2 Diagramme de cas d'utilisation

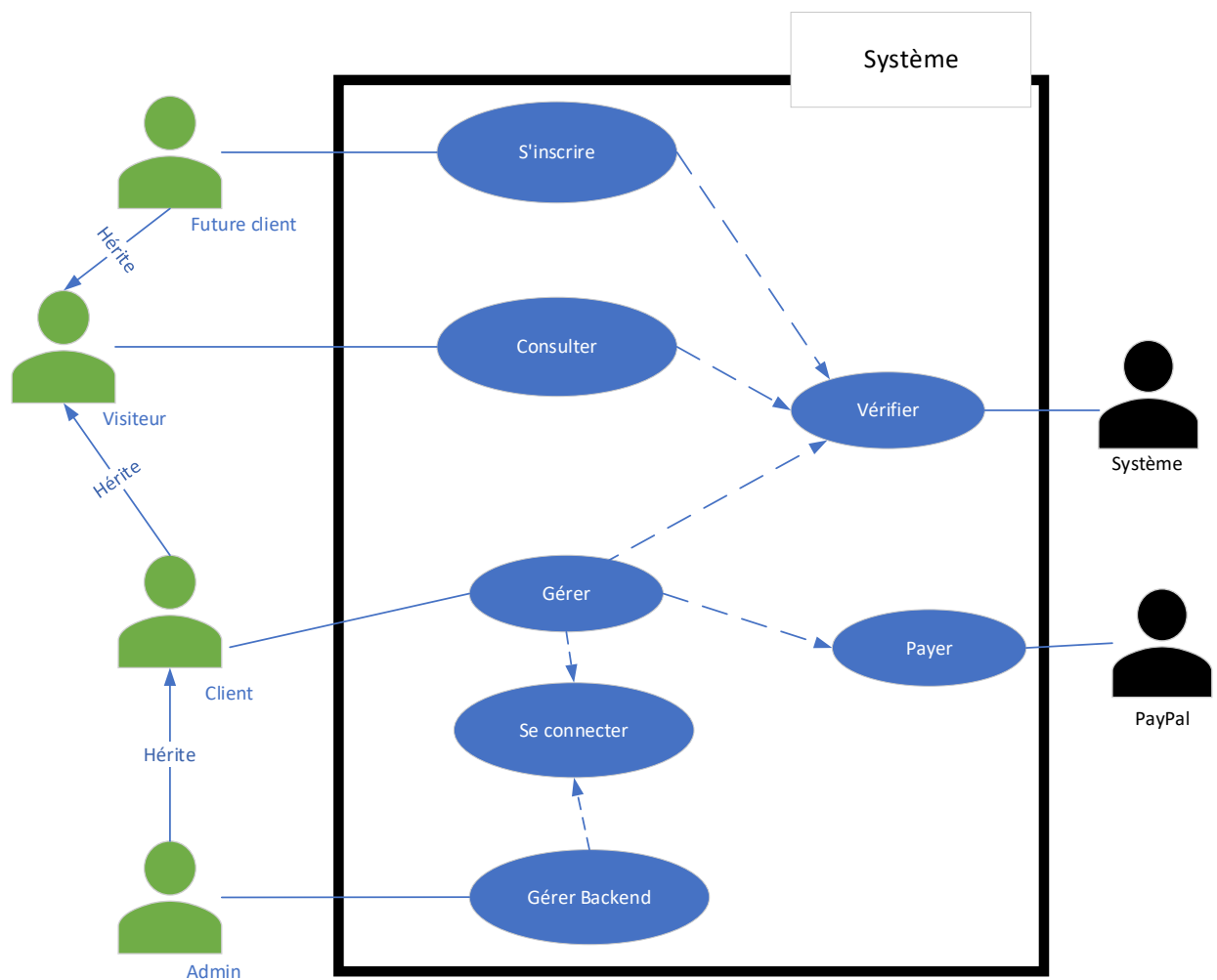


FIGURE 12 SCHEMA VISIO : DIAGRAMME DE CAS D'UTILISATION

5.3 Diagramme de navigation - Consulter

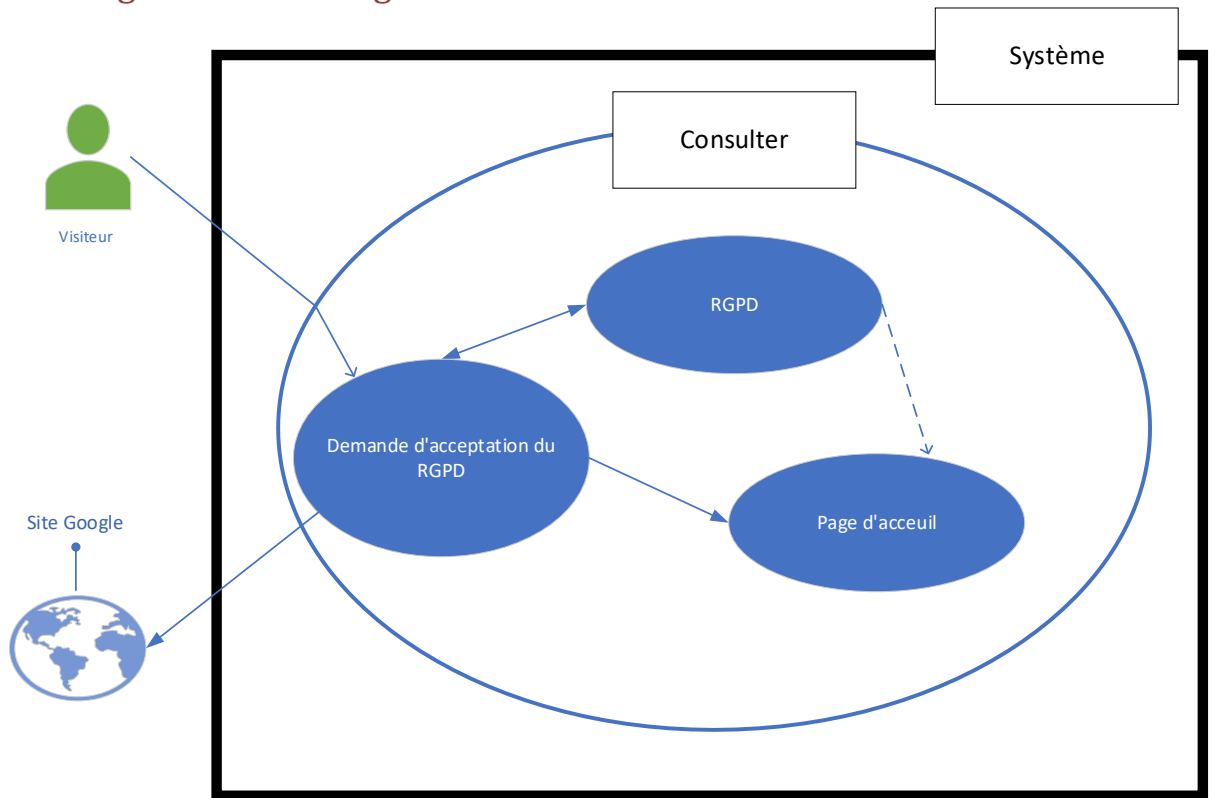


FIGURE 13 SCHEMA VISIO : DIAGRAMME DE NAVIGATION - CONSULTER

5.3.1 Cas d'utilisation : Consulter

Nom	Consulter
Description	<p>Lors de son arrivée sur l'application, l'utilisateur à un rôle de visiteur, il devra avant toute chose accepter les termes et conditions du RGPD (Règlement Général sur la Protection des Données) de l'Union européenne.</p> <p>S'il accepte, il peut consulter la page d'accueil.</p> <p>S'il ne l'accepte pas, il sera redirigé vers la page d'accueil de Google.</p>
Acteur	Visiteurs
Préconditions	<ol style="list-style-type: none"> 1. Le système doit être en mode Online. 2. Le visiteur accède à l'application via l'adresse http.
Scénario A	<ol style="list-style-type: none"> 1. Le système affiche la page d'accueil mais sans permettre d'action. 2. Le système affiche un popup de demande d'acceptation du RGPD. 3. Le visiteur lit le RGPD et accepte les conditions. 4. Le système ferme le popup de demande d'acceptation du RGPD. 5. Le système permet désormais des actions sur la page d'accueil. 6. Le visiteur peut : <ol style="list-style-type: none"> a. Consulter les biens mis en ligne. b. Trier sa recherche par : <ol style="list-style-type: none"> i. Type de bien. ii. Province et ville.
Scénario alternatif	<p>Le cas commence après le scénario A.2.</p> <p>Le visiteur n'accepte pas les conditions du RGPD en cliquant sur le bouton : « <i>je refuse</i> ».</p> <ul style="list-style-type: none"> • Le système ferme le popup de demande d'acceptation du RGPD. • Le système ferme l'application et redirige le visiteur vers une page Google.
Postcondition	Le visiteur peut consulter les biens mis en ligne.

5.4 Diagramme de navigation - S'inscrire

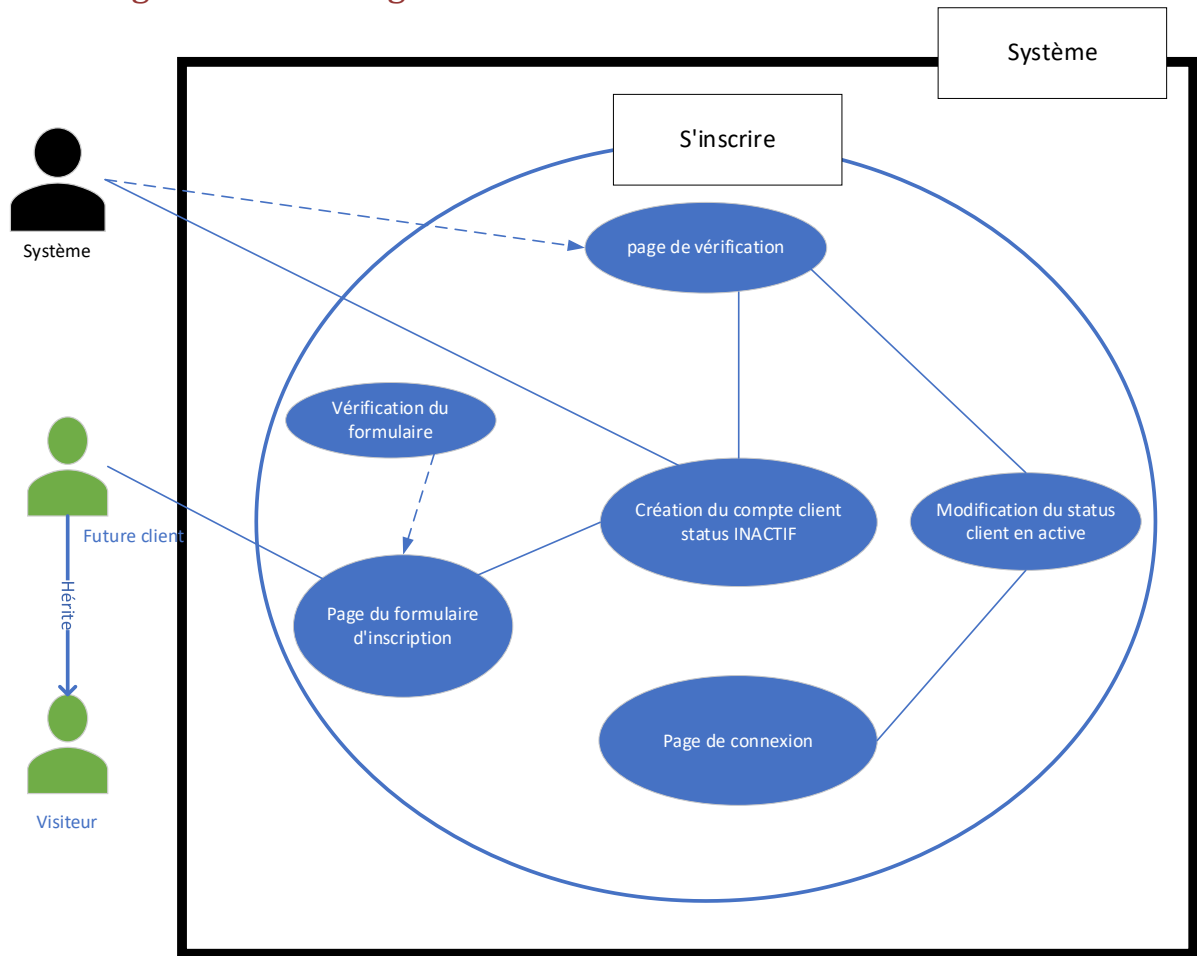


FIGURE 14 SCHEMA VISIO : DIAGRAMME DE NAVIGATION - S'INSCRIRE

5.4.1 Cas d'utilisation : S'inscrire

Nom	S'inscrire.
Description	Le futur client est un visiteur qui désire devenir client de Mobembo.cd. Ce désir est identifié lorsque le visiteur clique sur le bouton : « <i>Inscription</i> ».
Acteur	Futur client.
Précondition	Effectuer le scénario A de « Consulter ».
Scénario A	<ol style="list-style-type: none"> 1. Le système affiche le formulaire d'inscription. 2. Le futur client complète le formulaire d'inscription. 3. Le système vérifie le formulaire. 4. Le futur client soumet le formulaire au système. 5. Le système crée un compte client statut inactif. 6. Le futur client devient client. 7. Le système envoi un e-mail contenant un code de validation à la boîte mail du client. 8. Le système affiche le formulaire de vérification du code de validation. 9. Le client entre le code de validation. 10. Le système vérifie le code de validation. 11. Le client peut se connecter.
Scénario alternatif A	<p>Le cas commence au scénario A.3 Le futur client essaye d'envoyer un formulaire vide, incomplet ou incohérent.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système n'affiche pas le bouton d'envoi. 3. Le futur client complète ou modifie le formulaire. 4. Retour au scénario A.3
Scénario alternatif B	<p>Le cas commence au scénario A.10. Le client essaye d'envoyer un formulaire de validation vide.</p> <ol style="list-style-type: none"> 1. Le système n'affiche pas le bouton d'envoi. 2. Le client ne peut pas se connecter. <p>Le client fourni un code de validation incorrecte.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le client ne peut pas se connecter. <p>Le client entre un code de validation correcte. Retour au scénario A.11</p>
Postconditions	<ol style="list-style-type: none"> 1. Le futur client devient client. 2. Le client peut se connecter.

5.5 Diagramme de cas d'utilisation client

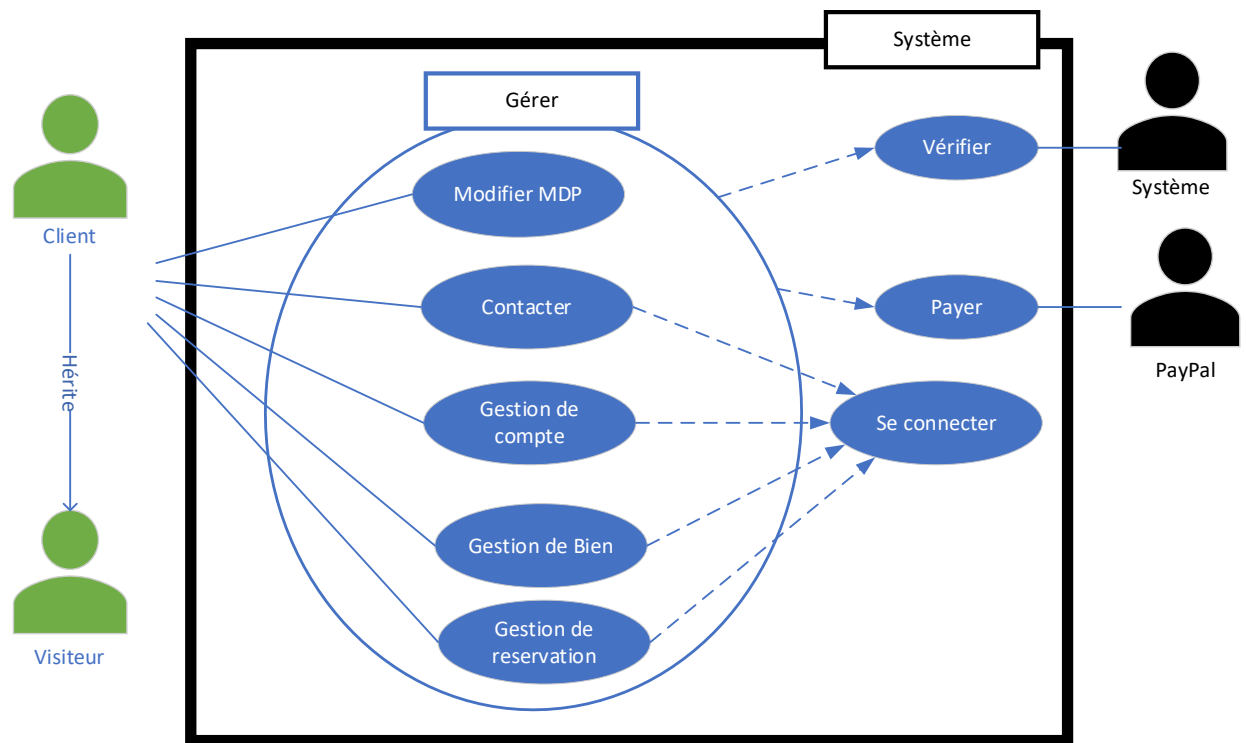


FIGURE 15 SCHEMA VISIO : DIAGRAMME DE CAS D'UTILISATION CLIENT

Le client est une personne physique. Il est reconnu par l'application en tant que détenteur d'un compte actif ou non. Il peut se connecter, ou si son compte est en monde inactif, il peut l'activer ou modifier son mot de passe.

S'il est connecté, il peut gérer les réservations, les biens, son compte et contacter l'administrateur de Mobembo.cd, à partir de l'application.

5.5.1 Diagramme de navigation client – Se connecter

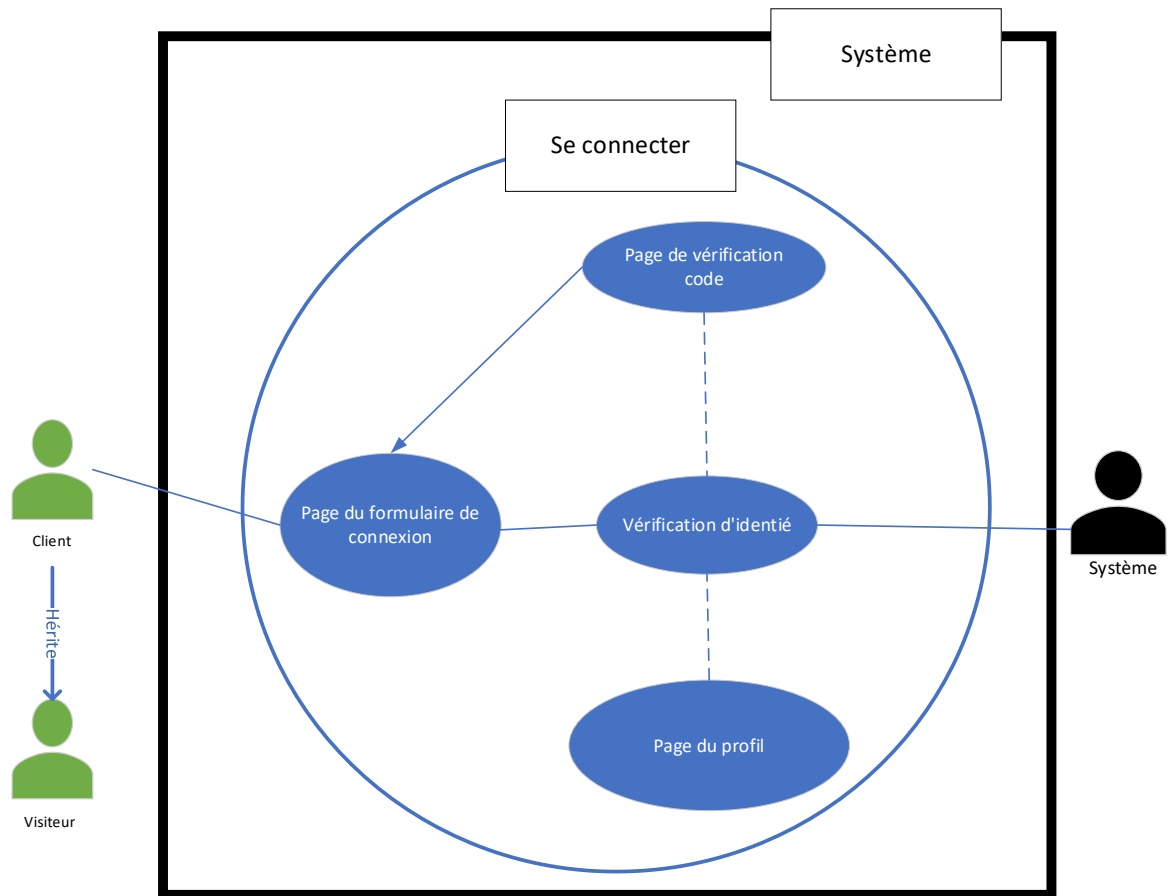


FIGURE 16 SCHEMA VISIO : DIAGRAMME DE NAVIGATION CLIENT - SE CONNECTER

5.5.1.1 Cas d'utilisation : Se connecter

Nom	Se connecter.
Description	Le cas commence lorsque le client clique sur le bouton : « connexion ».
Acteur	Client.
Préconditions	<ol style="list-style-type: none"> Être inscrit. Avoir un compte actif.
Scénario A	<ol style="list-style-type: none"> Le système affiche le formulaire de connexion. Le client remplit le formulaire de connexion et le soumet. Le système vérifie l'identité. Le client est connecté. Le système affiche la page du profil.
Scénario alternatif	<p>Le cas commence au scénario A.3.</p> <p>Le client renvoi un formulaire vide ou incorrect.</p> <ol style="list-style-type: none"> Le système affiche un message d'erreur. Le client n'est pas connecté. <p>Retour au scénario A.2.</p>
Postconditions	Le client est connecté.

5.5.2 Diagramme de navigation client - Modification du mot de passe

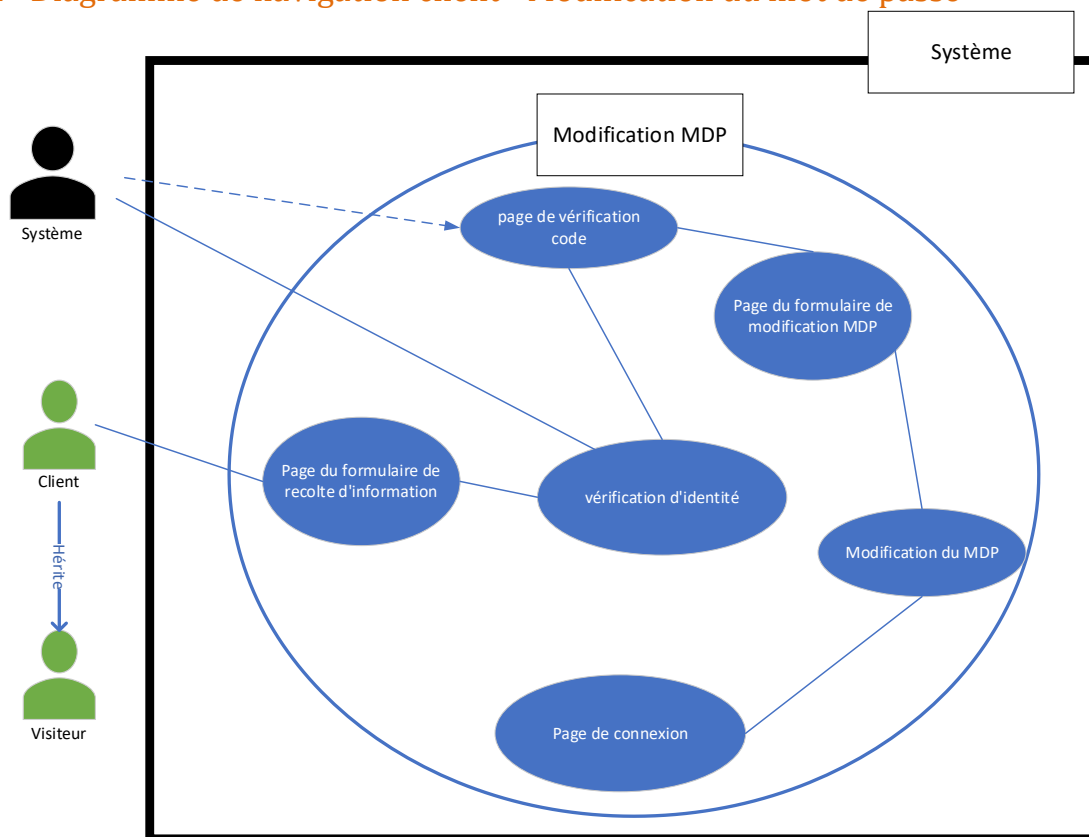


FIGURE 17 SCHEMA VISIO : DIAGRAMME DE NAVIGATION CLIENT - MODIFICATION MOT-DE-PASSE

5.5.2.1 Cas d'utilisation : Modification du mot de passe

Nom	Modification du mot de passe.
Description	Le client ne se rappelle plus son mot de passe, ou désire le changer. Le cas commence lorsque le client clique sur le bouton : « <i>Mot de passe oublié ?</i> »
Acteur	Client.
Préconditions	Avoir un compte.
Scénario A	<ol style="list-style-type: none"> 1. Le système affiche le formulaire de modification du mot de passe. 2. Le client remplit et soumet le formulaire. 3. Le système vérifie le formulaire. 4. Le système modifie le mot de passe du client. 5. Le système affiche la page de connexion.
Scénario alternatif	<p>Le cas commence au scénario A.3.</p> <p>Le client a soumis un formulaire vide ou un formulaire qui ne permet pas d'identifier le client.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système ne change pas le mot de passe. <p>Retour au scénario A.2.</p>
Postconditions	Mot de passe modifié.

5.5.3 Diagramme de navigation client - Gestion de compte

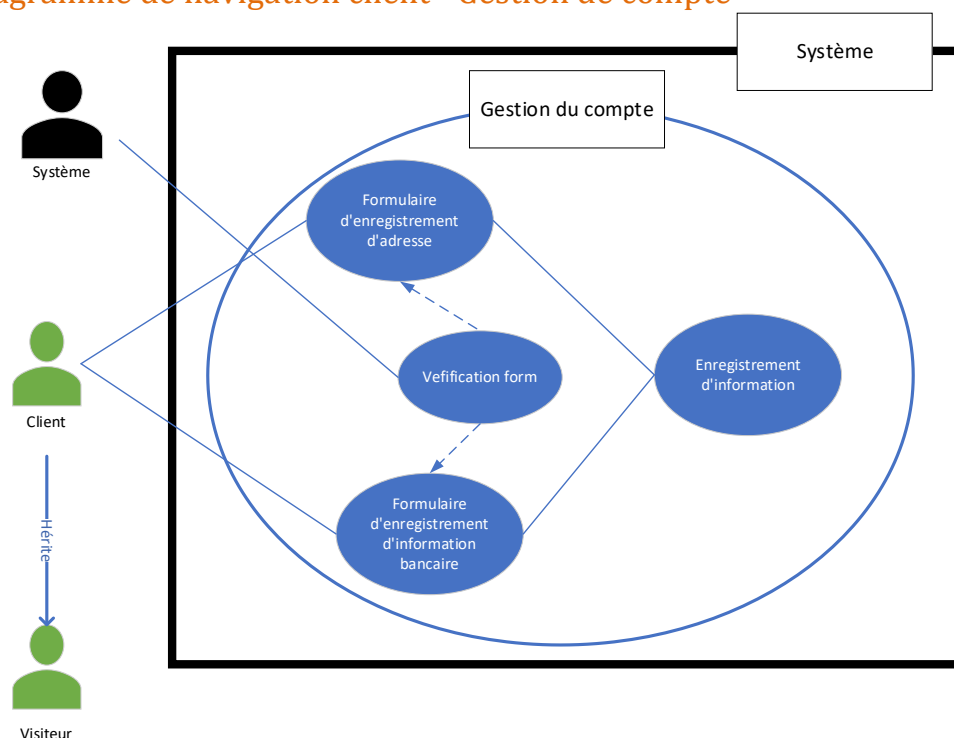


FIGURE 18 SCHEMA VISIO : DIAGRAMME DE NAVIGATION CLIENT - GESTION DE COMPTE

5.5.3.1 Cas d'utilisation : Enregistrement d'adresse

Nom	Enregistrement d'adresse.
Description	<p>Pour effectuer des réservations ou mettre un bien en ligne, le système a besoin d'informations bancaires et connaître l'adresse du client afin d'établir le contrat.</p> <p>Le cas commence lorsque le client clique sur le bouton : « <i>Profil du client : nom prénom</i> » à droite dans le menu principal.</p>
Acteur	Client.
Préconditions	Être connecté.
Scénario A	<ol style="list-style-type: none"> 1. Le système affiche le formulaire d'enregistrement d'adresse et le formulaire d'enregistrement d'informations bancaires. 2. Le client remplit et soumet le formulaire d'enregistrement d'adresse. 3. Le système vérifie le formulaire. 4. Le système enregistre l'adresse du client. 5. Le système ferme le formulaire d'enregistrement d'adresse. 6. Le système affiche les premières lettres de l'adresse du client.
Scénario alternatif	<p>Le cas commence au scénario A.3.</p> <p>Le client soumet un formulaire vide ou ayant un champ vide.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur. 2. Le système n'enregistre pas le formulaire. <p>Retour au scénario A.2.</p>
Postconditions	Adresse privée ajoutée.

5.5.3.2 Cas d'utilisation : Enregistrement d'informations bancaires

Nom	Enregistrement d'informations bancaires.
Description	<p>Pour effectuer des réservations ou mettre un bien en ligne, le système a besoin des informations bancaires et de l'adresse du client afin d'établir le contrat.</p> <p>Le cas commence lorsque le client clique sur le bouton : « <i>Profil du client : nomEtPrenom</i> » à droite dans le menu principal.</p>
Acteur	Client.
Préconditions	Être connecté.
Scénario A	<ol style="list-style-type: none">1. Le système affiche le formulaire d'enregistrement d'adresse et le formulaire d'enregistrement des informations bancaires.2. Le client remplit et soumet le formulaire d'enregistrement des informations bancaires.3. Le système vérifie le formulaire.4. Le système enregistre les informations bancaires du client.5. Le système ferme le formulaire d'enregistrement des informations bancaires.6. Le système affiche les premières lettres des informations bancaires du client.
Scénario alternatif	<p>Le cas commence au scénario A.3.</p> <p>Le client soumet un formulaire vide ou ayant un champ vide.</p> <ol style="list-style-type: none">1. Le système affiche un message d'erreur.2. Le système n'enregistre pas le formulaire. <p>Retour au scénario A.2.</p>
Postconditions	Informations bancaires ajoutée.

5.5.4 Diagramme de navigation client - Gestion de Bien - Création

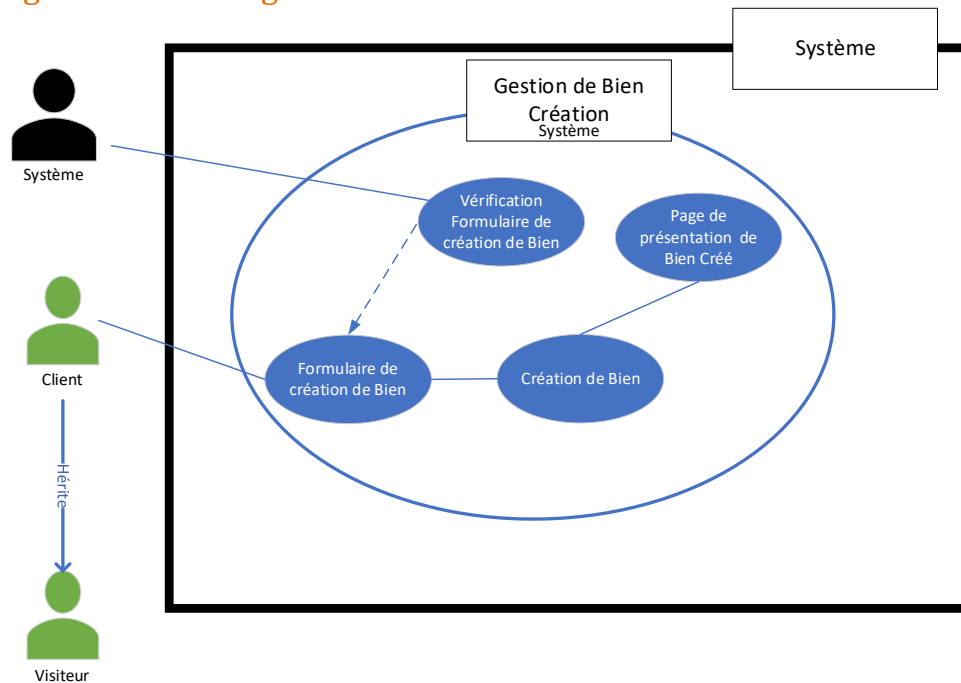


FIGURE 19 SCHEMA VISIO : DIAGRAMME DE NAVIGATION CLIENT - GESTION DE BIEN - CREATION

5.5.4.1 Cas d'utilisation gestion de bien : Création

Nom	Gestion de bien : Création.
Description	Le cas commence lorsque le client clique sur le bouton : « Créer un bien ».
Acteur	Client.
Préconditions	1. Avoir un compte actif. 2. Être connecté.
Scénario A	1. Le système affiche le formulaire de création de bien. 2. Le client remplit le formulaire. 3. Le système vérifie le formulaire. 4. Le système affiche le bouton « Je crée mon bien ». 5. Le client soumet le formulaire en cliquant sur le bouton « Je crée mon bien ». 6. Le système crée le bien avec un mode actif à « false ». a. Le bien est désormais disponible dans la page « Mes biens ». 7. Le système affiche la page « Mes biens ».
Scénario alternatif	Le cas commence au scénario A.3. Le client soumet un formulaire vide ou un formulaire avec un ou plusieurs champs vides. 1. Le système n'affiche pas le bouton « Je crée mon bien ». 2. Le système affiche un message d'erreur. Retour au scénario A.2.
Postconditions	Bien disponible dans la page « Mes biens ».

5.5.5 Diagramme de navigation client - Gestion de Bien - Suppression

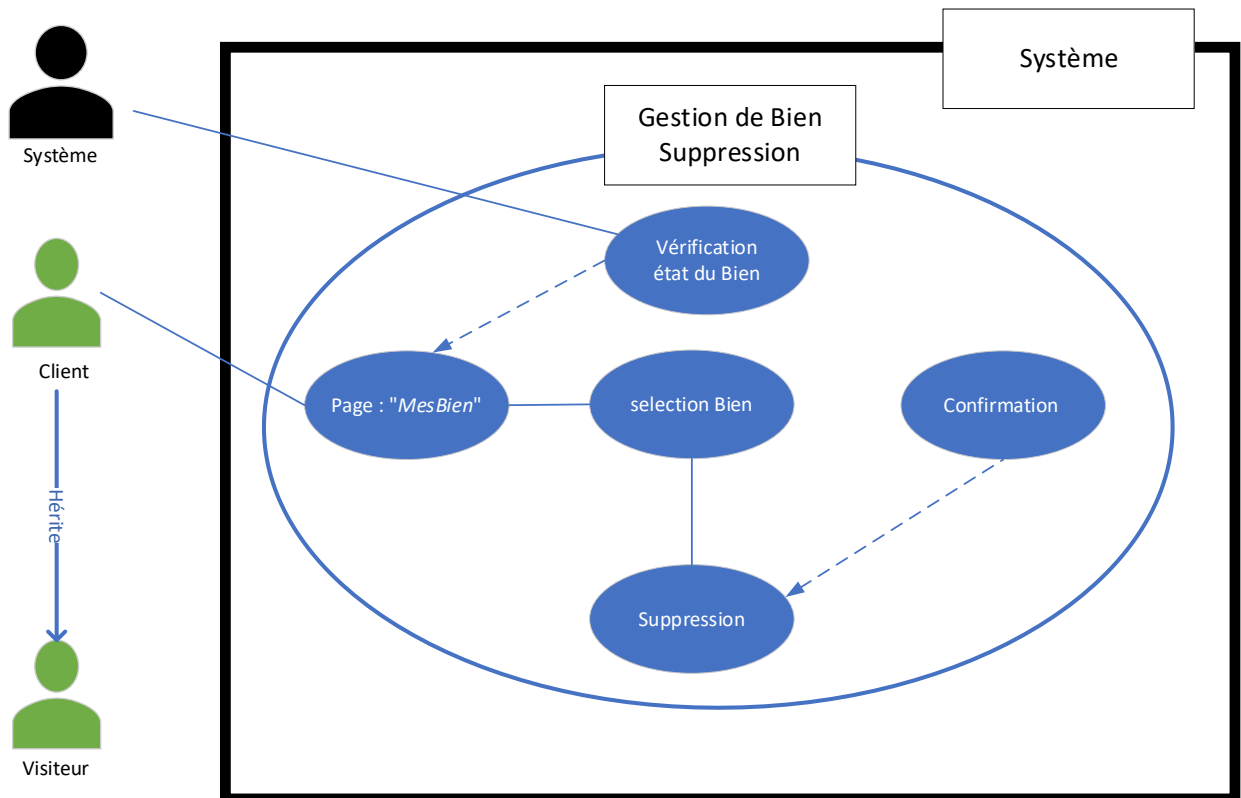


FIGURE 20 SCHEMA VISIO : DIAGRAMME DE NAVIGATION CLIENT - GESTION DE BIEN - SUPPRESSION

5.5.5.1 Cas d'utilisation gestion de bien : Suppression

Nom	Gestion de bien : Suppression.
Description	Le client désire supprimer le bien de sa liste. Le cas commence lorsque le client clique sur le bouton : « <i>Mes Biens</i> ».
Acteur	Client.
Préconditions	<ol style="list-style-type: none"> 1. Avoir un compte actif. 2. Être connecté. 3. Avoir un bien qui n'est pas lié à un contrat actif.
Scénario A	<ol style="list-style-type: none"> 1. Le système affiche la page « Mes biens ». 2. Le système fait des vérifications. 3. Le client sélectionne le bien à supprimer en cliquant sur l'icône « supprimer ». 4. Le système affiche un message d'avertissement. 5. Le client consent à l'avertissement en cliquant sur le bouton « supprimer ». 6. Le système génère une phrase de validation et l'affiche, en demandant au client de la retranscrire.

	<p>7. Le client retranscrit la phrase de validation et clique sur le bouton « je confirme la suppression ».</p> <p>8. Le système vérifie la phrase entrée par le client.</p> <p>9. Le système supprime le bien.</p>
Scénario alternatif A	<p>Le cas commence au scénario A.2.</p> <p>Si le client n'a créé aucun bien ou que le bien à supprimer est lié à un contrat de mis en ligne actif, aucune action ne sera possible.</p> <p>Fin du processus.</p>
Scénario alternatif B	<p>Le cas commence au scénario A.8.</p> <p>Si la phrase que le client a soumise ne correspond pas à la phrase générée par le système :</p> <p>Le système affiche un message d'erreur.</p> <p>Retour au scénario A.7.</p>
Postconditions	Suppression du bien.

5.5.6 Diagramme de navigation client - Gestion de Bien - Mis en ligne

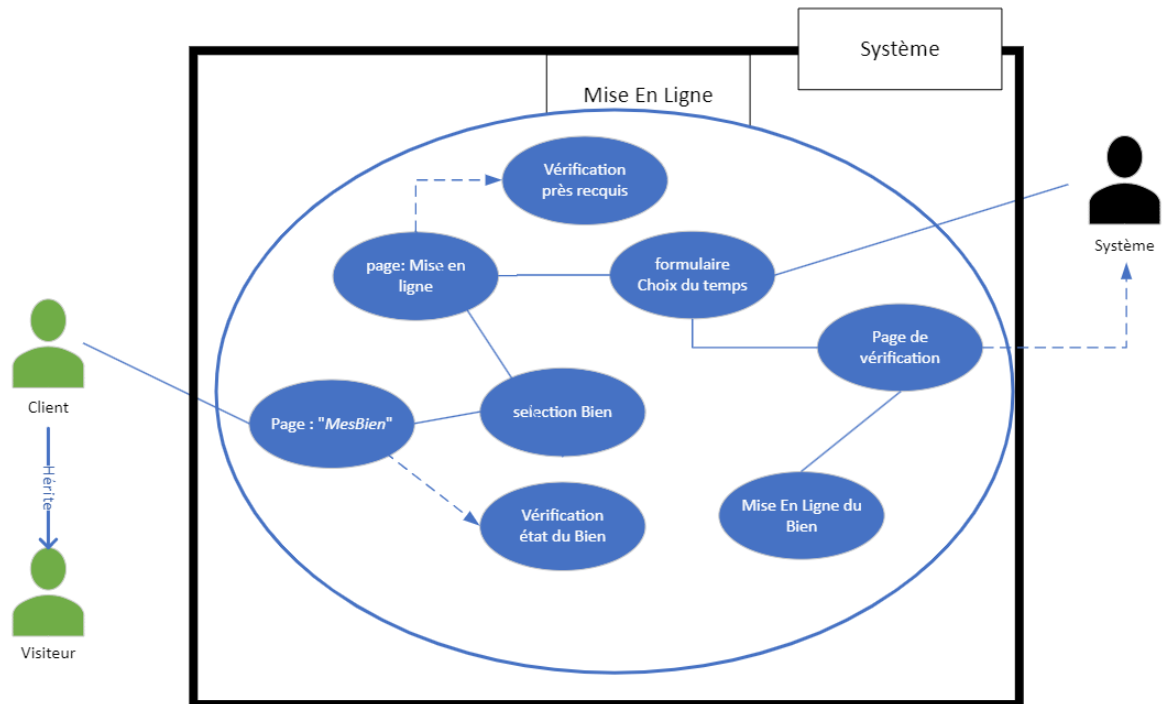


FIGURE 21 SCHEMA VISIO : DIAGRAMME DE NAVIGATION CLIENT - GESTION DE BIEN - MIS EN LIGNE

5.5.6.1 Cas d'utilisation gestion de bien : Mis en ligne

Nom	Gestion de bien : Mis en ligne.
Description	Le client voudrait que son bien soit visible par tous, et qu'il puisse faire l'objet de réservations. Le cas commence lorsque le client clique sur le bouton : <i>"Mes Biens"</i> .
Acteur	Client.
Préconditions	<ol style="list-style-type: none"> 1. Avoir un compte actif. 2. Être connecté. 3. Avoir un bien qui n'est pas lié à un contrat actif. 4. Avoir enregistré une adresse privée. 5. Avoir enregistré des informations bancaires.
Scénario A	<ol style="list-style-type: none"> 1. Le système affiche la page « Mes biens ». 2. Le système fait des vérifications. 3. Le client sélectionne le bien à mettre en ligne en cliquant sur le bouton « Mettre en ligne ». 4. Le système affiche la page « Mis en ligne ». 5. Le système fait des vérifications.

	6. Le client coche la case devant la phrase « J'accepte les termes et les conditions du contrat de mis en ligne ». 7. Le système affiche un formulaire demandant au client de sélectionner la durée de mis en ligne. 8. Le client sélectionne une durée et clique sur le bouton « Mettre en ligne ». 9. Le système envoi un e-mail contenant un code de validation à la boite mail du client. 10. Le système affiche la page de vérification de code de validation. 11. Le client soumet le formulaire avec le code de validation reçu sur sa boite mail. 12. Le système fait des vérifications. 13. Le système crée le contrat de mis en ligne entre le client et TSHISHI-RDC représenté par l'administrateur du site. 14. Le système met le bien en ligne (visible par tous). 15. Le bien peut être réservé. 16. Le client peut consulter le contrat de mis en ligne.
Scénario alternatif A	Le cas commence au scénario A.2. Si le client n'a créé aucun bien ou que le bien à mettre en ligne est lié à un contrat de mis en ligne actif, aucune action ne sera possible. Fin du processus.
Scénario alternatif B	Le cas commence au scénario A.5. Si le client n'a pas enregistré d'informations bancaires ou d'adresse privée : 1. Le système affiche un message invitant le client à compléter son profil. 2. Impossible de mettre en ligne. 3. Le client n'a qu'un seul choix, cliquer sur le bouton « Annuler ».
Scénario alternatif C	Le cas commence au scénario A.12. Si le client soumet le formulaire vide ou avec un code de validation incorrect : 1. Le système affiche un message d'erreur. 2. Le client peut : a. Reprendre au scénario A.11. b. Reprendre au scénario A.1 en cliquant sur le bouton « Annuler ».
Postconditions	Mis en ligne du bien (visible par tous). Le bien peut être réservé.

5.5.7 Diagramme de navigation client - Gestion de Bien - Contrat mis en ligne

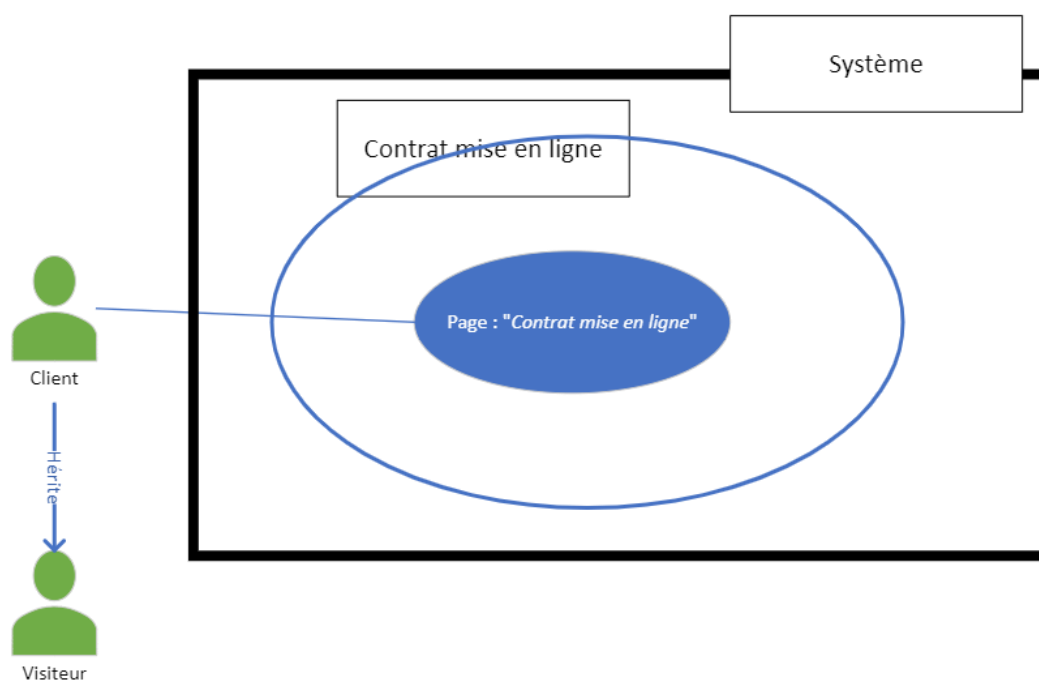


FIGURE 22 SCHEMA VISIO : DIAGRAMME DE NAVIGATION CLIENT - GESTION DE BIEN - CONTRAT MIS EN LIGNE

5.5.7.1 Cas d'utilisation gestion de bien : Contrat mis en ligne

Nom	Gestion des biens : Contrat mis en ligne.
Description	Le client désire voir tous les contrats actifs ou non de mis en ligne. Le cas commence lorsque le client clique sur le bouton : <i>"Contrat mis en ligne"</i> .
Acteur	Client.
Préconditions	1. Avoir un compte actif. 2. Être connecté. 3. Avoir un bien qui est lié à un contrat de mis en ligne actif ou inactif.
Scénario A	1. Le système fait des vérifications. 2. Le système affiche la liste des contrats de mis en ligne du client. Cette liste est triée du contrat le plus récent au plus ancien.
Scénario alternatif	Le cas commence au scénario A.1. Si le client ne possède aucun contrat de mis en ligne, le système affiche un message disant que la liste est vide.
Postconditions	Le client peut consulter ses contrats de mis en ligne.

5.5.8 Diagramme de navigation client - Gestion de Bien – Annulation contrat mis en ligne

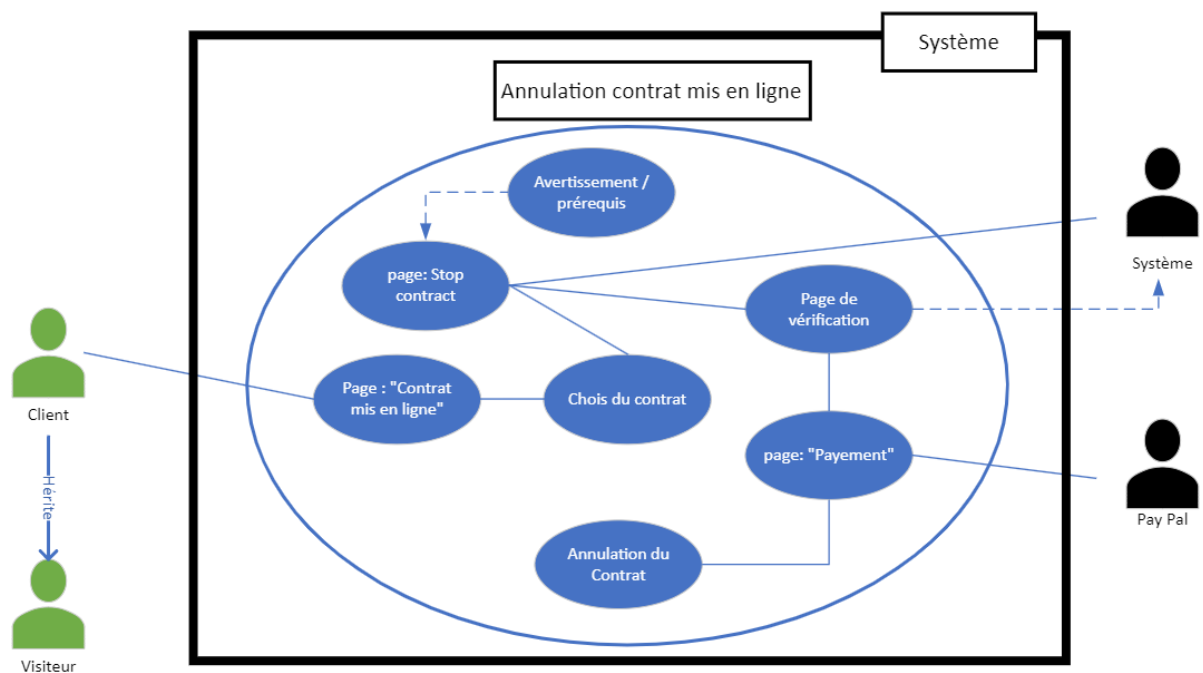


FIGURE 23 SCHEMA VISIO : DIAGRAMME DE NAVIGATION CLIENT - GESTION DE BIEN - ANNULATION CONTRAT MIS EN LIGNE

5.5.8.1 Cas d'utilisation gestion de bien : Annulation contrat mis en ligne

Nom	Gestion de bien : Annulation contrat mis en ligne.
Description	Le client désire mettre fin au contrat de mis en ligne, pour diverses raisons. Le cas commence lorsque le client clique sur le bouton : <i>“Contrat mis en ligne”</i> .
Acteur	Client.
Préconditions	<ol style="list-style-type: none"> 1. Avoir un compte actif. 2. Être connecté. 3. Avoir un bien qui est lié à un contrat de mis en ligne actif.
Scénario A	<ol style="list-style-type: none"> 1. Le système fait des vérifications. 2. Le système affiche la liste des contrats de mis en ligne du client. Cette liste est triée du plus récent au plus ancien. 3. Le client choisit le contrat à annuler en cliquant sur le bouton « Mettre fin au contrat ». 4. Le système affiche la page « Annulation contrat mis en ligne » avec un message d'avertissement et une case à cocher devant la phrase « J'accepte et je poursuis l'arrêt de mon contrat ». 5. Le client coche la case devant la phrase « J'accepte et je poursuis l'arrêt de mon contrat ».

	<p>6. Le système affiche un message d'avertissement sur fond rouge reprenant les informations de l'action en cours.</p> <p>7. Le client clique sur le bouton « Continuer ».</p> <p>8. Le système génère une phrase et l'affiche.</p> <p>9. Le système demande au client de retranscrire la phrase générée.</p> <p>10. Le client retranscrit la phrase générée par le système.</p> <p>11. Le système fait des vérifications.</p> <p>12. Le système affiche le bouton « Je confirme la demande d'arrêt du contrat ».</p> <p>13. Le client clique sur le bouton « Je confirme la demande d'arrêt du contrat ».</p> <p>14. Le système envoie un e-mail au client contenant un code de validation.</p> <p>15. Le système affiche la page de vérification de code de validation.</p> <p>16. Le client remplit le formulaire avec le code de validation reçu par e-mail et le soumet.</p> <p>17. Le système fait des vérifications.</p> <p>18. Le système affiche la page de paiement.</p> <p>19. Le client paye.</p> <p>20. Le système fait des vérifications.</p> <p>21. Le système modifie la date de fin du contrat mis en ligne par la date du jour/J, et modifie l'état du contrat mis en ligne, ainsi que du bien associé en mode inactif.</p> <p>22. Le bien n'est plus en ligne (plus visible par tous).</p> <p>23. Le client n'est plus tenu par un contrat pour le bien souhaité.</p>
Scénario alternatif A	<p>Le cas commence au scénario A.1.</p> <p>Si le client ne possède aucun contrat de mis en ligne actif.</p> <p>Pas d'annulation de contrat mis en ligne.</p>
Scénario alternatif B	<p>Le cas commence au scénario A.11.</p> <p>Si le client ne retranscrit pas la phrase à l'identique que celle générée par le système, le système n'affiche pas le bouton « je confirme la demande d'arrêt du contrat ».</p> <p>Pas d'annulation du contrat mis en ligne.</p>
Scénario alternatif C	<p>Le cas commence au scénario A.17.</p> <p>Si le client remplit le formulaire avec un code de validation incorrect, le système affiche un message d'erreur.</p> <p>Pas d'annulation du contrat mis en ligne.</p>
Scénario alternatif D	<p>Le cas commence au scénario A.20.</p> <p>Si le client ne finit pas le processus de paiement ou que le processus de paiement est un échec, le système affiche un message d'erreur.</p> <p>Pas d'annulation de contrat mis en ligne.</p>
Postconditions	<p>Le client n'est plus tenu par un contrat pour le bien souhaité.</p> <p>Le bien n'est plus en ligne.</p>

5.5.9 Diagramme de navigation client - Gestion de Bien – Information sur le Bien.

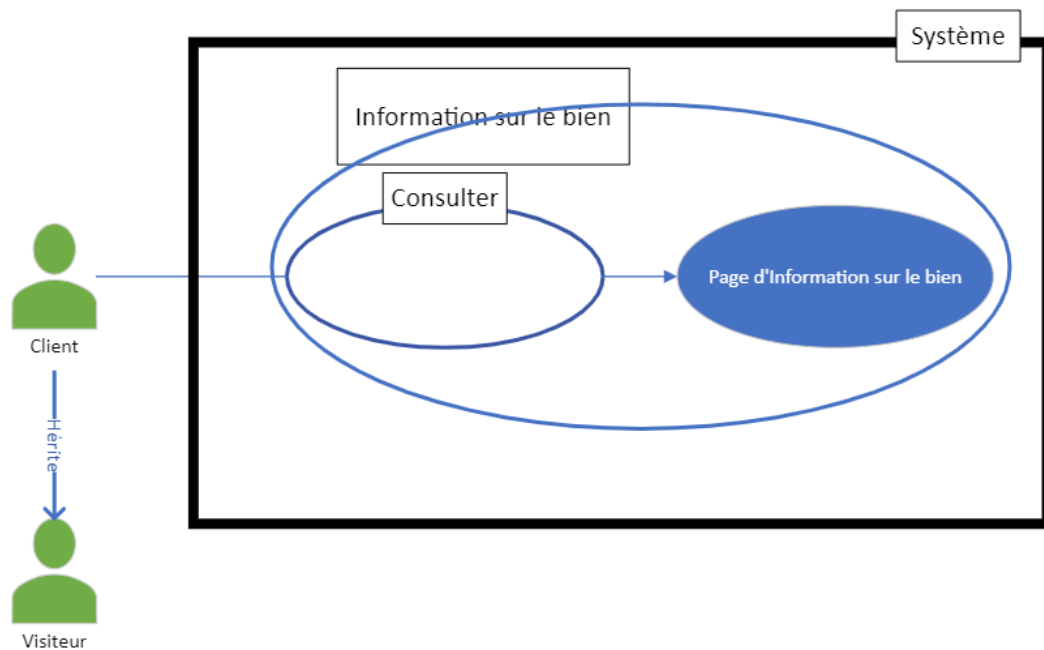


FIGURE 24 SCHEMA VISIO : DIAGRAMME DE NAVIGATION CLIENT - GESTION DE BIEN - INFORMATION SUR LE BIEN

5.5.9.1 Cas d'utilisation gestion de bien : Information sur le bien

Nom	Gestion de bien : information sur le bien.
Description	Le client aimerait de plus amples informations que celles présentes dans l'affichage simplifiée. Le cas commence lorsque le client clique sur un bien mis en ligne.
Acteur	Client.
Préconditions	1. Avoir un compte actif. 2. Être connecté.
Scénario A	1. Le client clique sur un bien mis en ligne. 2. Le système affiche la page « Information sur le bien ».
Scénario alternatif	
Postconditions	Avoir de plus amples informations sur un bien.

5.5.10 Diagramme de navigation client - Gestion de Bien – « Liker » un Bien.

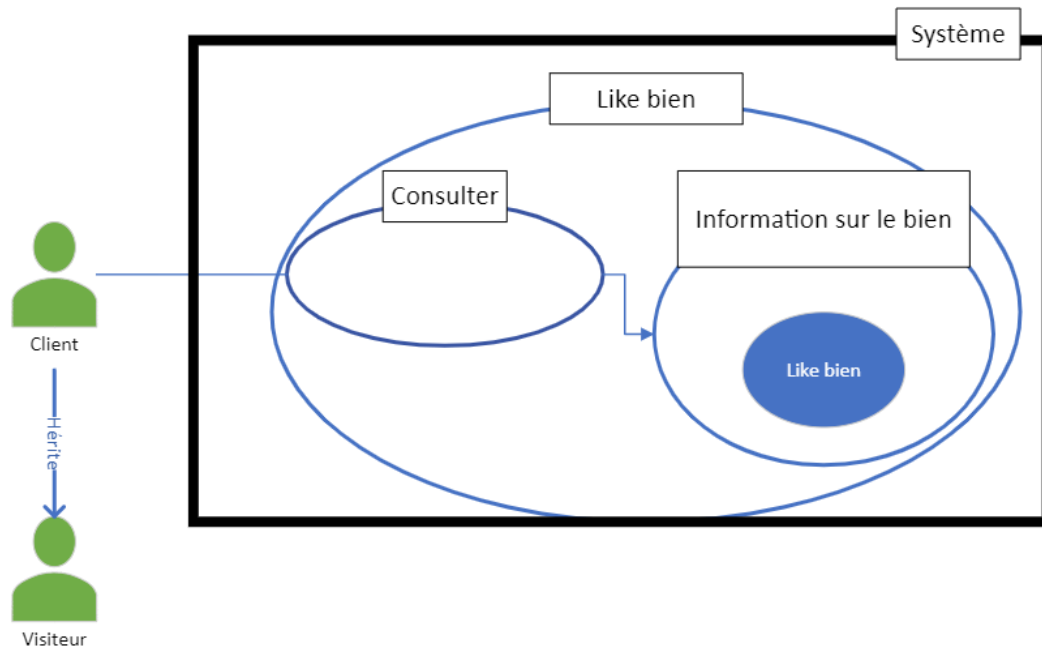


FIGURE 25 SCHEMA VISIO : DIAGRAMME DE NAVIGATION CLIENT - GESTION DE BIEN - LIKE BIEN

5.5.10.1 Cas d'utilisation gestion de bien : Like bien

Nom	Gestion de bien : Like bien.
Description	Le client désire laisser une note, positive ou négative, sur un bien. Le cas commence lorsque le client clique sur un bien mis en ligne.
Acteur	Client.
Préconditions	Informations sur le bien.
Scénario A	<ol style="list-style-type: none"> 1. Le client clique sur le bouton bleu avec l'icône « yeux en cœur ». 2. Le système affiche un message : « Merci pour votre réaction ». 3. Le système ajoute un « Like » au bien. 4. Si le client avait déjà cliqué sur le bouton bleu avec l'icône « yeux en cœur ». 5. Le système affiche un message : « Merci pour votre réaction ». 6. Le système retire un « Like » au bien.
Scénario alternatif	
Postconditions	Ajouter ou retirer un « Like ».

5.5.11 Diagramme de navigation client - Gestion de réservation – Création

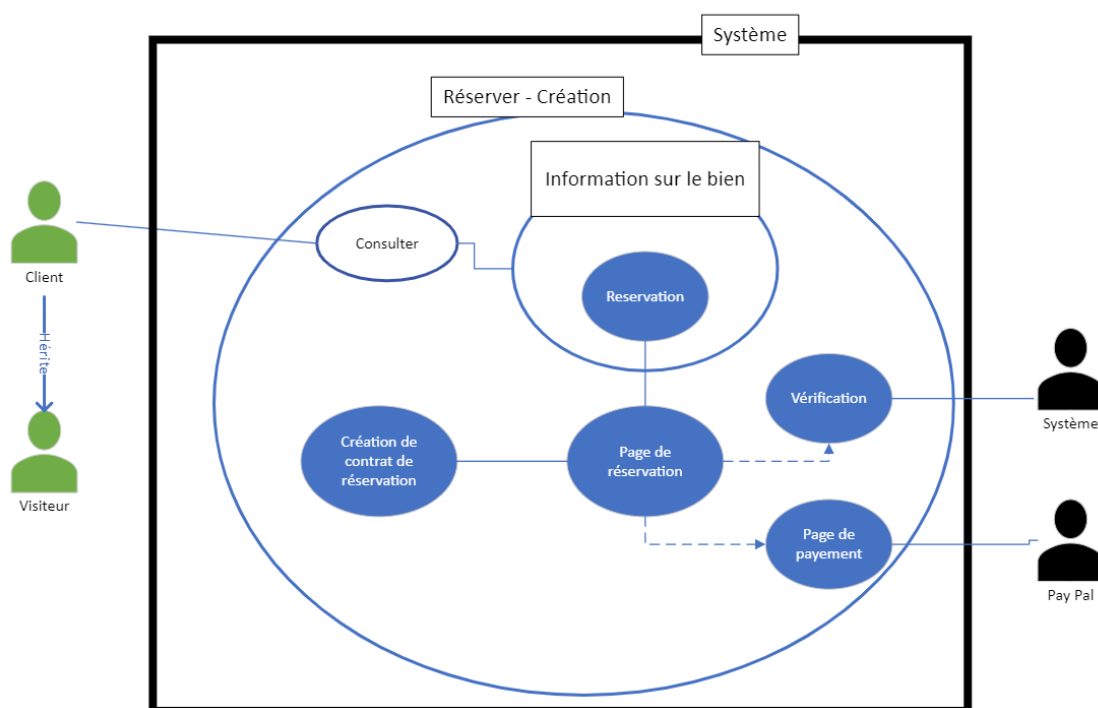


FIGURE 26 SCHEMA VISIO : DIAGRAMME DE NAVIGATION CLIENT - GESTION DE RESERVATION - CREATION

5.5.11.1 Cas d'utilisation gestion de réservation : Création

Nom	Gestion de réservation : Création.
Description	Le cas commence lorsque le client clique sur un bien mis en ligne.
Acteur	Client.
Préconditions	<ol style="list-style-type: none"> 1. Cas d'utilisation gestion de bien : Information sur le bien. 2. Avoir enregistré une adresse privée. 3. Avoir enregistré des informations bancaires.
Scénario A	<ol style="list-style-type: none"> 1. Le client clique sur le bouton « Réservation ». 2. Le système fait des vérifications. 3. Le système affiche la page de réservation avec un message d'avertissement, concernant l'action en cours, ainsi qu'une case à cocher devant la phrase « J'accepte les termes et conditions du Contrat ». 4. Le client coche la case devant la phrase « J'accepte les termes et conditions du Contrat ». 5. Le système affiche un formulaire de réservation. 6. Le client remplit et soumet le formulaire de réservation. 7. Le système fait des vérifications. 8. Le système affiche le bouton « Passer au paiement ». 9. Le client clique sur le bouton « Passer au paiement ».

	<p>10. Le système envoie un e-mail au client, contenant un code de vérification.</p> <p>11. Le système affiche la page de vérification du code de validation.</p> <p>12. Le client soumet le formulaire avec le code de validation reçu par e-mail.</p> <p>13. Le système fait des vérifications.</p> <p>14. Le système affiche un récapitulatif de la réservation, le prix total à payer, ainsi qu'un bouton « Confirmer et payer ».</p> <p>15. Le client clique sur le bouton « Confirmer et payer ».</p> <p>16. Le système affiche la page de paiement.</p> <p>17. Le client paye.</p> <p>18. Le système fait des vérifications.</p> <p>19. Le système crée un contrat de réservation de bien entre le client et TSHISHI-RDC représenté par l'administrateur.</p> <p>20. Le bien est réservé par le client.</p>
Scénario alternatif A	<p>Le cas commence au scénario A.2.</p> <p>Si le client n'a pas enregistré d'informations bancaires ou d'adresse privée :</p> <ol style="list-style-type: none"> 1. Le système affiche un message invitant le client à compléter son profil. 2. Impossible de réserver. 3. Le client n'a qu'un seul choix, cliquer sur le bouton « Annuler ».
Scénario alternatif B	<p>Le cas commence au scénario A.7.</p> <p>Si le client soumet un formulaire vide ou avec un ou plusieurs champs vides ou avec une date d'arrivée antérieure à la date du jour ou avec une date de départ antérieure à la date d'arrivée, le système affiche un message d'erreur.</p> <p>Pas de réservation possible.</p>
Scénario alternatif C	<p>Le cas commence au scénario A.13.</p> <p>Si le client remplit le formulaire avec un code de validation incorrect, le système affiche un message d'erreur.</p> <p>Pas de réservation possible.</p>
Scénario alternatif D	<p>Le cas commence au scénario A.18.</p> <p>Si le client ne finit pas le processus de paiement ou que le processus de paiement est un échec, le système affiche un message d'erreur.</p> <p>Pas de réservation possible.</p>
Postconditions	Bien réservé par le client.

5.6 Diagramme de cas d'utilisation admin

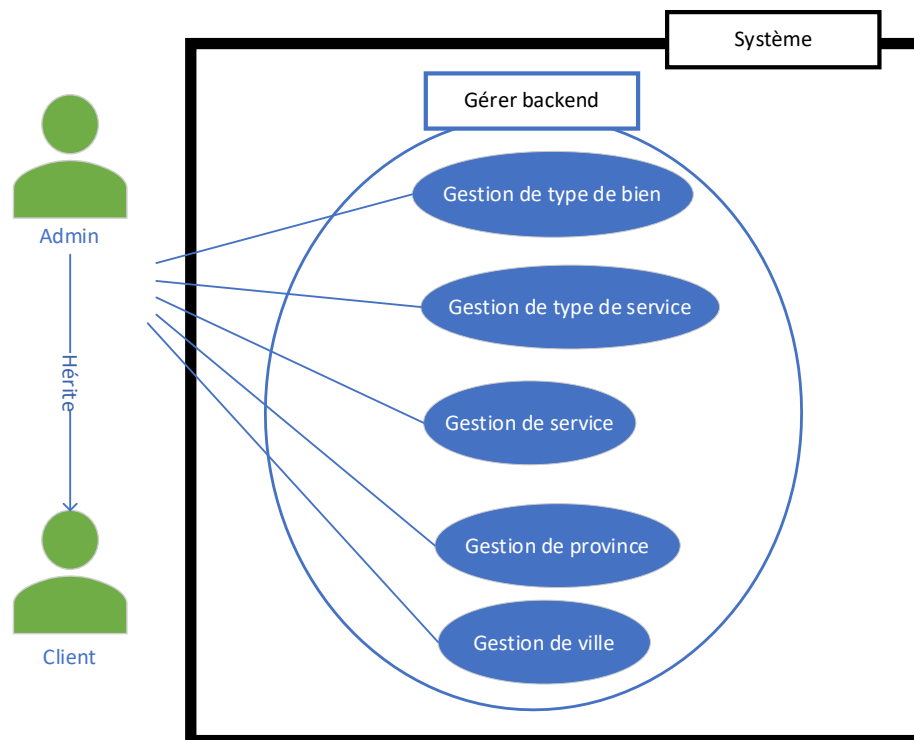


FIGURE 27 DIAGRAMME DE CAS D'UTILISATION ADMIN

5.6.1 Création du rôle administrateur

Nom	Création administrateur.
Description	Le cas commence lorsque le développeur reçoit l'ordre de créer un profil administrateur.
Acteur	Développeur.
Préconditions	Cas d'utilisation : inscription
Scénario A	<ol style="list-style-type: none"> 1. Le développeur se rend dans la base de données et modifie la table « personne » en donnant à la colonne « role_id » la valeur « 1 ». 2. Le système enregistre la modification. 3. Le client devient administrateur.
Scénario alternatif	
Postconditions	Le client devient administrateur.

5.7 Conclusion

Dans ce chapitre, grâce à la mise en place des diagrammes et des cas d'utilisation correspondant, la programmation s'est avérée plus simple et rapide.

6 Développement :

6.1 Introduction

Dans ce chapitre, je vais expliquer les cadres et les outils utilisés pour le développement de l'application.

Vous y retrouverez l'environnement de travail, les langages utilisés, les conseils d'installation locale.

J'aborderai également la description de la conception du backend et je terminerai avec les outils et mécanismes mis en place dans le cadre de la cybersécurité.

6.2 Environnement de travail

6.2.1 IntelliJ (6)

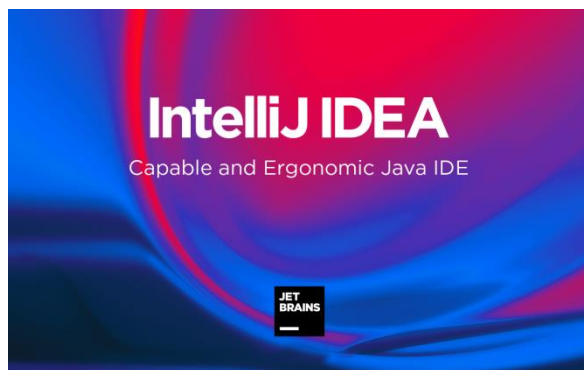


FIGURE 28 IMAGE ISSU DE GOOGLE - INTELLIJ IDEA

Pour la réalisation de cette application, j'ai utilisé comme IDE (Environnement De Développement), IntelliJ ultimate de JetBrains.

Elle amène, à mon sens, une aide considérable lors de la création de Rest-API Java, d'application Angular et bien d'autres. ⁶

Grâce à mon inscription en tant qu'étudiant en bachelier en informatique de gestion, j'ai pu acquérir une licence chez JetBrains.

Cette licence me donne accès à tous les outils de travail dans leur version complète que propose l'éditeur de logiciels durant toute la durée de mon cursus.

⁶ JetBrains. IntelliJ IDEA. *jetbrains.com*. [En ligne] 2022. <https://www.jetbrains.com/idea/promo/>.

6.2.2 WampServer



FIGURE 29 IMAGE ISSU DE GOOGLE - WAMPSEVER

(7)

Bien évidemment, il est possible de créer ou de faire fonctionner cette application avec d'autres environnements.

Pour le fonctionnement des scripts server, j'ai utilisé la plateforme de développement local de type WAMP, qui est WampServer. Cette plateforme est totalement gratuite.

Cet environnement comprend trois serveurs, un interpréteur de script, ainsi que phpMyAdmin pour l'administration Web des bases MySQL.⁷

⁷ Bourdon, Romain. WampServer. *wampserver.com*. [En ligne] <https://www.wampserver.com/>.

6.3 Langage utilisé

6.3.1 SQL

(8)

Pour la gestion d'une base de données, j'ai opté pour une base de données MySQL. Ce produit de l'entreprise Oracle, est fourni lors de l'installation de WampServer.⁸

Pour mon projet, j'ai utilisé la version 5.1.1 de PHPMYAdmin.



FIGURE 30 IMAGE ISSU DE GOOGLE - MYSQL

6.3.2 Java

(9)

Pour la réalisation de mon backend, j'ai opté pour la technologie de développement de l'entreprise Oracle.

Java est un langage très puissant et très verbeux.

Ce qui permet la réalisation de grand projet tout en gardant un total contrôle de chaque processus.⁹

Pour mon projet, j'ai utilisé la version Java 11 dans un environnement Spring Boot version 2.4.1-SNAPSHOT.



FIGURE 31 IMAGE ISSU DE GOOGLE - JAVA / SPRING

6.3.3 Type script

(10)

Pour la réalisation du frontend, j'ai opté pour le Framework open source basé sur du Type Script (une dérivation de JavaScript) qui est Angular.

Sa logique de séparation par module (Component) permet la réalisation d'application dynamique et très performante.¹⁰



FIGURE 32 IMAGE ISSU DE GOOGLE - ANGULAR

Pour mon projet, j'ai opté pour la version Angular Client 13.

⁸ Oracle. MySQL. *mysql.com*. [En ligne] 2022. <https://www.mysql.com/fr/>.

⁹ —. Java. *dev.java*. [En ligne] 2021. <https://dev.java/>.

¹⁰ Gougple. Angular. *angular.io*. [En ligne] 2010-2022. <https://angular.io/>.

6.4 Installation et mise en route en local.

Tout le code nécessaire au bon fonctionnement de l'application peut être retrouvé sur GitHub. A l'adresse : <https://github.com/tshishi3687/>

- Le Frontend (client Web) :
 - Nom : frond-end-projet-RDC
 - Emplacement : tshishi3687/frond-end-projet-RDC
 - Git Clone : <https://github.com/tshishi3687/frond-end-projet-RDC.git>
- Le Backend (REST-API) :
 - Nom : rdc_touristique
 - Emplacement : tshishi3687/rdc_touristique
 - Git Clone : https://github.com/tshishi3687/rdc_touristique.git

6.4.1 Base de données

La base de données est de type MySQL. Pour une utilisation de l'application en local, il est nécessaire d'installer un server local comme WampServer ou XampServer contenant une version de PHPMyAdmin de minimum 5.1.1.

Le code SQL fourni contient tout le nécessaire pour le bon fonctionnement de l'application.

Ce code peut être récupéré de deux façons.

1. Si vous possédez déjà le code du backend, vous retrouverez le code SQL dans le dossier `data_access/sql/mobembo.cd.sql`
2. Sinon via Git :
[rdc_touristique/src/main/java/com/example/rdc_touristique/data_access/sql/](#)

6.4.2 Rest-API Java

Il est nécessaire d'avoir ou d'obtenir un éditeur de code Java tel que IntelliJ, NetBeans, ou Jdeveloper.

Il faut s'assurer, après le clone, que toutes les dépendances sont installées.

La plupart des éditeurs le font automatiquement après un Clone Git.

Il faut s'assurer aussi que la base de données porte le même nom que celle déclarée dans le fichier « application-properties ».

Le REST-API répond sur le port 8081.

6.4.3 Client Web

Le Git Clone permet à lui seul l'installation du Angular Client ainsi que l'Angular Projet.

Comme pour Le REST-API, les dépendances s'installeront automatiquement lors du Clone.

6.5 Le Backend

6.5.1 Base de données

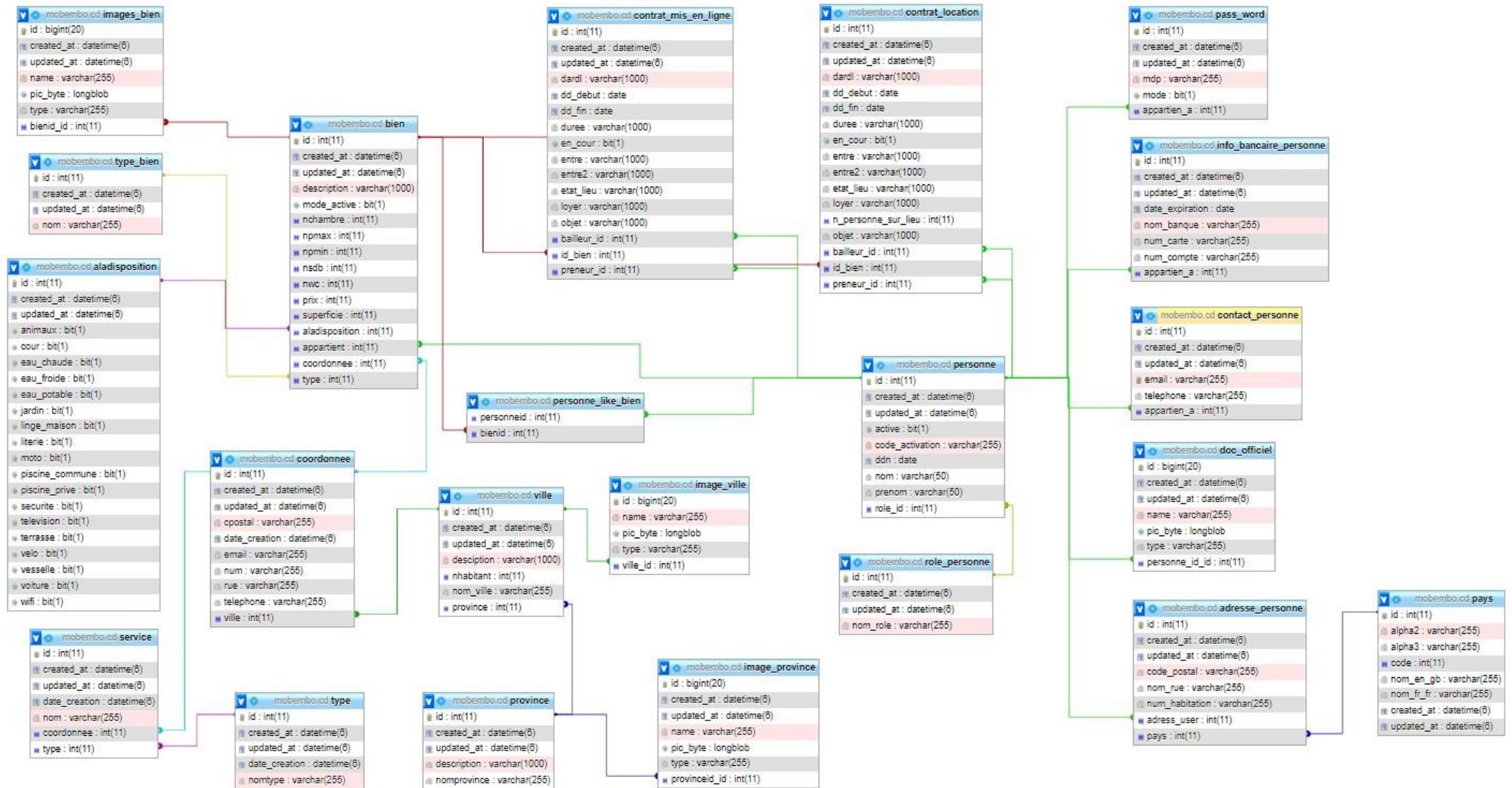


FIGURE 33 IMAGE ISSU DE MYSQL - SCHEMA BASE DE DONNEES MOBEMBO.CD

6.5.1.1 Base de données : introduction

La base de données est la partie la plus importante de mon application, mais également celle qui demande le moins de temps.

En effet, mis à part la création de la base de données et la création d'un événement, qui seront ajoutés à la fin de la création de l'application, toute la création des tables et des colonnes, ainsi que la mise en place des contraintes relationnelles, sont gérées par la partie REST-API, grâce à l'environnement Spring Boot.

6.5.1.2 Base de données : Événement

Nom	retrait_contrat_et_bien_en_ligne	UPDATE bien JOIN contrat_mis_en_ligne ON bien.id = contrat_mis_en_ligne.id_bien SET bien.mode_active = 0, contrat_mis_en_ligne.en_cour = 0 WHERE contrat_mis_en_ligne.dd_fin < NOW()
État	Enabled	
Type d'événement	Recurring	
Exécuter chaque	Toutes les minutes	

Cet événement a été mis en place après la création de toutes les tables.

- Son but est de garantir l'automatisation des processus courants.
Elle garantit que chaque bien mis en ligne (visible par tous) est relié à un contrat actif et dont la date de fin n'est pas antérieure à la date courante.

6.5.1.3 Base de données : Liste des tables

Les tables		
• AdressePersonne	• Aladisposition	• ContactPersonne
• ContratLocation	• ContratMisEnLigne	• Coordonnee
• Doc_Officiel	• ImageBien	• ImageProvince
• ImageVille	• InforbancairePersonne	• PassWord
• Pays	• Personne	• Province
• Role_Personne	• Service	• Type_Service
• Type_Bien	• Ville	• Bien

6.5.1.4 Base de données : Relations entre tables

6.5.1.4.1 *Personne – Doc_Officiel*

Une personne peut avoir zéro ou plusieurs documents officiels.
Un document officiel appartient à une personne.



6.5.1.4.2 *Personne – RolePersonne*

Une personne possède un rôle.
Un rôle peut être lié à zéro ou plusieurs personnes.



6.5.1.4.3 *Personne – Adresse_Personne*

Une personne peut avoir zéro ou une adresse personnelle.
Une adresse personnelle appartient à une personne.



6.5.1.4.4 *Personne – Contact_personne*

Une personne possède un contact personnel.
Un contact personnel appartient à une personne.



6.5.1.4.5 *Personne – PassWord*

Une personne possède un mot de passe.
Un mot de passe appartient à une personne.



6.5.1.4.6 *Personne – Info_Bancaire_personne*

Une personne peut avoir zéro ou une information bancaire personnelle.
Une information bancaire appartient à une personne.



6.5.1.4.7 *Personne – Contrat_mis_en_ligne*

Une personne peut être liée à zéro ou plusieurs contrats de mis en ligne.
Un contrat de mis en ligne appartient à une personne.



6.5.1.4.8 *Personne – Contrat_Location*

Une personne peut être liée à zéro ou plusieurs contrats de location.
Un contrat de location appartient à une personne.



6.5.1.4.9 *Personne – Bien*

Une personne peut être propriétaire de zéro ou plusieurs biens.
Un bien appartient à une personne.



6.5.1.4.10 Personne – Bien (Like)

Une personne peut Liker (aimer) zéro ou plusieurs biens.
Un bien peut être Liké (aimé) par zéro ou plusieurs personnes.



6.5.1.4.11 Bien – Image_bien

Une bien peut avoir zéro ou plusieurs images.
Une image appartient à un bien.



6.5.1.4.12 Bien – Type_Bien

Un bien possède un type.
Un type de bien peut être lié à zéro ou plusieurs biens.



6.5.1.4.13 Bien – Aladisposition

Un bien possède une mise à disposition (Aladisposition).
Une mise à disposition appartient à un bien.



6.5.1.4.14 Bien - Coordonnee

Un bien possède une coordonnée.
Une coordonnée est liée à zéro ou plusieurs biens.



6.5.1.4.15 Coordonnee – Ville

Une coordonnée est associée à une ville.
Une ville peut être associée à zéro ou plusieurs coordonnées.



6.5.1.4.16 Service – Coordonnee

Un service est associé à une coordonnée.

Une coordonnée peut être associée à zéro ou plusieurs services.



6.5.1.4.17 Service – Type_Service

Un service possède un type.

Un type de service peut être lié à zéro ou plusieurs services.



6.5.1.4.18 Ville – Image_Ville

Une ville peut posséder zéro ou plusieurs Images.

Une image appartient à une ville.



6.5.1.4.19 Ville – Province

Une ville appartient à une province.

Une province peut être associée à zéro ou plusieurs villes.



6.5.1.4.20 Province – Image_Province

Une province peut avoir zéro ou plusieurs Images.

Une Image appartient à une province.



6.5.1.4.21 Adresse_Personne – Pays

Une adresse personnelle est liée à un pays.

Un pays peut être associé à zéro ou plusieurs adresses personnelles.



6.5.2 REST-API

6.5.2.1 Diagramme de navigation des données

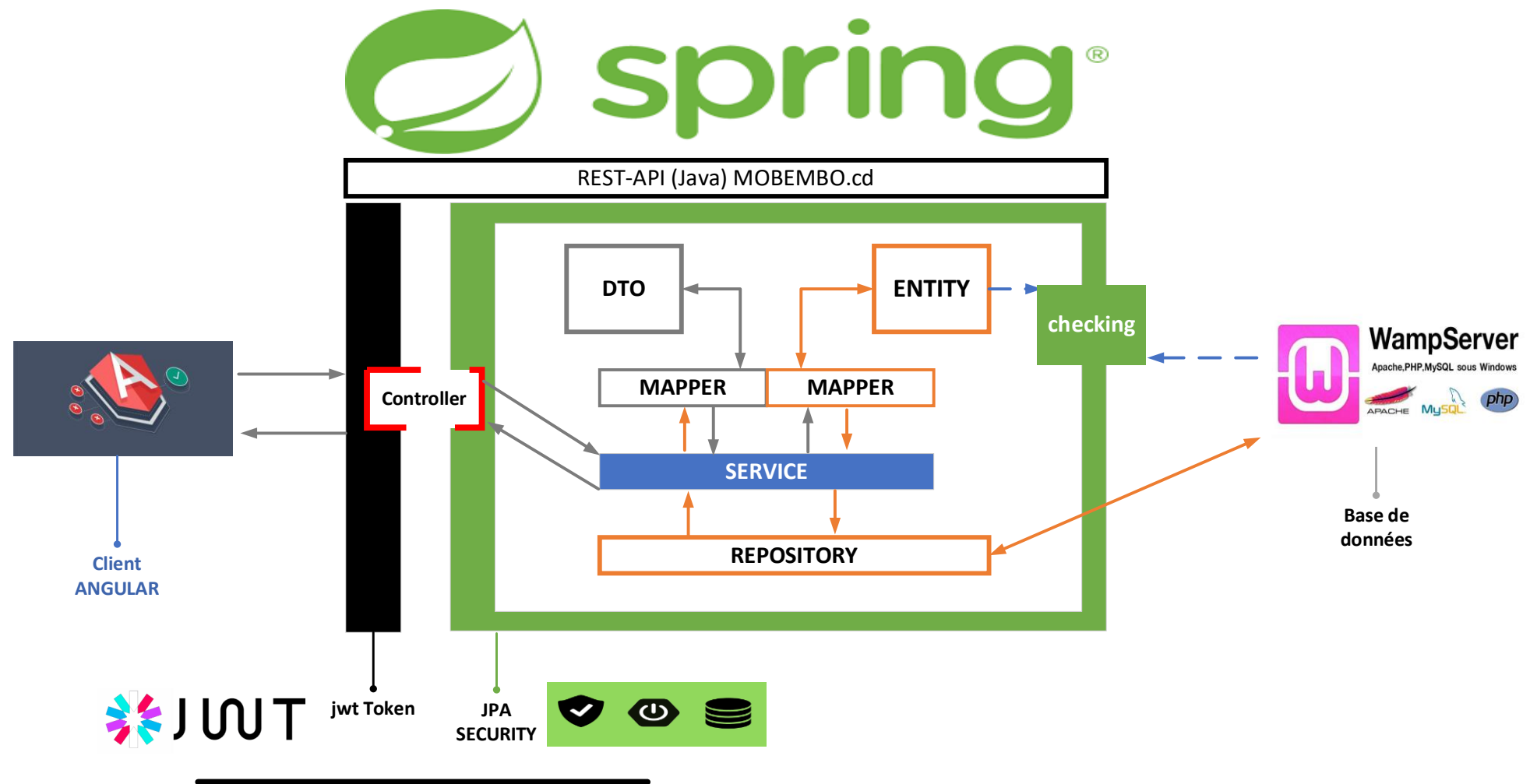
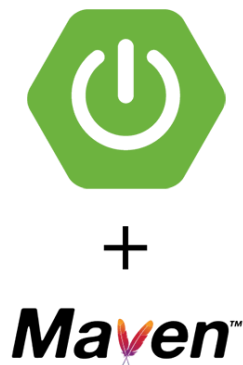


FIGURE 34 SCHEMA VISIO - DIAGRAMME DE NAVIGATION DES DONNEES PROJET SPRING BOOT

6.5.2.2 Maven



Pour la réalisation du backend, j'ai choisi de créer un projet Maven.

Ce dernier met en place un environnement développement spring boot en POO (Programmation Orienté Objet).

Ceci permet une meilleure valorisation du code dû à la possibilité de mettre en place une architecture du projet final plus cohérente.

FIGURE 35 IMAGE ISSU DE GOOGLE - MAVEN / SPRING

J'ai implémenté cet environnement avec différentes dépendances (bibliothèques) qui me

permettent d'intégrer certaines méthodes.

Pour voir toutes les dépendances utilisées, je vous invite à consulter le fichier « pom.xml » du projet.

6.5.2.3 Application Properties

Avant de commencer l'architecture du projet, il faut modifier le fichier « application.properties » en définissant quelques contraintes :

- Le chemin ainsi que les authentifications de la base de données.
- L'initialisation du schéma de la base de données.
- Le port de diffusion.

Comme suit :

```
spring.datasource.url jdbc:mysql://localhost:3306/mobembo.cd?serverTimezone=UTC
spring.datasource.username root
spring.datasource.password
```

```
spring.jpa.hibernate.ddl-auto validate
spring.jpa.generate-ddl true
spring.jpa.show-sql true
```

```
server.port 8081
```

Explication :

Le nom de ma base de données est « mobembo.cd », avec comme username « root » et password (vide).

`spring.jpa.hibernate.ddl-auto=validate` indique que lors de l'exécution du programme, la vérification de l'architecture de la base de données est comparée avec les entités déclarées dans mon application.

Si ce n'est pas le cas, il génère une exception et arrête l'application.

Je vous conseille, si vous copiez mon code, de le mettre en update

`spring.jpa.hibernate.ddl-auto update` et avant de lancer l'application, de créer une base de données avec le même nom indiqué ci-dessus.

Ce qui permettra la création de la base de données avec les bonnes contraintes.

6.5.2.4 Architecture

L'application est divisée en sept parties :

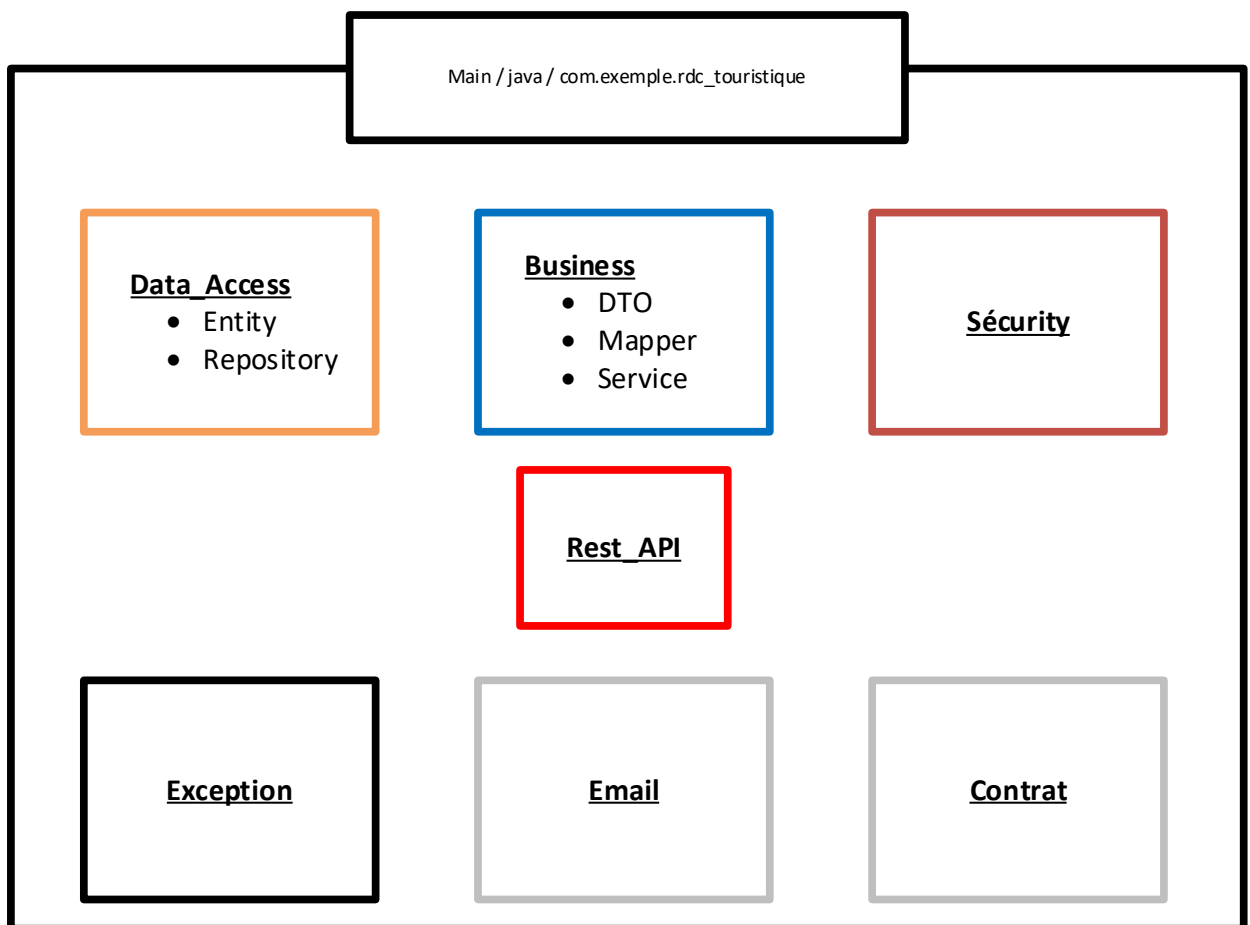


FIGURE 36 SCHEMA VISIO - PLAN BACKEND

6.5.2.4.1 Dossier Data_Access

Le dossier Data_Access est composé de classes et méthodes permettant d'interagir avec la base de données.

Deux sous dossier le composent : Entity et Repository.

6.5.2.4.1.1 Entity

(Fichier = nom de la table = Nomdelatable.java)

Le dossier Entity contient vingt classes qui représentent mes entités (les tables) ainsi que les contraintes de ma base de données.

Elles héritent toutes de la classe BaseEntity et j'utilise la librairie de Java Persistence ainsi que celle de LOMBOK.

6.5.2.4.1.1.1 La classe BaseEntity

L'abstract classe BaseEntity va intégrer deux colonnes de type Date lors de la modélisation d'une classe Entity dans la base de données.

La colonne createdAt et updatedAt.

- CreatedAt : Chaque fois qu'il y aura une insertion dans une table, cette colonne prendra la valeur, la date et l'heure du server.
- UpdatedAt : à chaque modification d'une ligne, cette colonne prendra la valeur, la date et l'heure du server.

6.5.2.4.1.1.2 Model Entity JPA

@Entity

@Entity est une annotation de JPA indiquant que cette CLASSE est la réplique exacte d'une table dans ma base de données.

@Id

@Id est une annotation de JPA indiquant que la variable qui suit est l'identifiant de ma table.

@GeneratedValue (strategy = GenerationType.IDENTITY)

@GeneratedValue est une annotation de JPA indiquant que la variable qui suit est unique et auto incrémentée.

@Column ou @Column (length = 1000)

@Column est une annotation de JPA indiquant que la variable qui suit est une colonne de ma table. Il est possible de spécifier les propriétés.

Exemple : length = 1000 représente varchar (1000).

@OneToMany

@OneToMany est une annotation de JPA indiquant que la variable qui suit est une association. Classe (table) a une relation 1-N.

@OneToOne

@OneToOne est une annotation de JPA indiquant que la variable qui suit est une association. Classe (table) a une relation 1-1.

@ManyToOne

@ManyToOne est une annotation de JPA indiquant que la variable qui suit est une association. Classe (table) a une relation N-1.

@ManyToMany

@ManyToMany est une annotation de JPA indiquant que la variable qui suit est une association. Classe (table) a une relation N-N.

6.5.2.4.1.1.3 Annotation Lombok

@Getter

@Getter est une annotation de Lombok indiquant que lors de l'appel de cette classe, il doit générer automatiquement les fonctions de récupération de chaque variable de la classe.

@Setter

@Setter est une annotation de Lombok indiquant que lors de l'appel de cette classe, il doit générer automatiquement les fonctions de modification de chaque variable de la classe.

@NoArgsConstructor

@NoArgsConstructor est une annotation de Lombok indiquant que cette classe peut être initialisée sans constructeur.

@AllArgsConstructor

@AllArgsConstructor est une annotation de Lombok indiquant que cette classe peut être initialisée avec constructeur.

6.5.2.4.1.2 Repository

(Fichier nom de la table plus repository = NomdelatableRepository.java)

Contient vingt interfaces, chacune de ces interfaces est liée à une classe portant le même nom de table.

Annotation : `extends JpaRepository` < classe liée, le type de l'identifiant de la classe liée >

Le type d'identifiant de la classe liée = `int id` = Integer.

Toutes les interfaces de Repository donnent accès au CRUD de base (Creat, ReadOne / ReadAll, Update et Delete).

Il est aussi possible d'implémenter une interface avec d'autres méthodes en fonction des besoins.

Par exemple pour ma classe (table) province, je sais que cette classe (table) contient des variables (colonnes), donc comme sur SQL, il est possible de faire tous les types de SELECT.

Avec `Liste<Province> findByOrderByNomprovinceAsc()` ; je crée une méthode dans mon interface ProvinceRepository qui me génère une liste de provinces triées par ordre alphabétique.

Cela revient à la même chose que de faire en SQL :

```
SELECT *  
FROM Province  
ORDER BY NomProvince Asc
```

6.5.2.4.2 Dossier Business

Le dossier Business est composé de classes et de méthodes permettant de traiter les demandes provenant de la partie Frontend.

Trois sous-dossiers le composent : DTO, Mapper et Service.

6.5.2.4.2.1 Dossier DTO

(Fichier = nom de la classe concernée plus l'annotation DTO = NomdelatableDTO.java)

Le dossier DTO contient trente-trois classes représentant la structure de chaque modèle d'objet accepté dans le body lors d'une requête ou une réponse HTTP.

J'utilise la librairie de LOMBOK.

Comme pour les Entités, ces classes ont les annotations suivantes :

@Getter

@Setter

@NoArgsConstructor

@AllArgsConstructor

Ainsi que @ToString

6.5.2.4.2.1.1 @ToString

@ToString est une annotation de Lombok qui crée une fonction, qui renvoie une donnée de type String correspondant aux informations MINIMALES de ladite classe.

Ce qui permet la lecture en console de la variable pour la vérification des données reçues ou envoyées.

Mise à part la classe CreatPersonneDTO, FavoryDTO, LikeBienDTO, MdpDTO et ModifPassDTO, toutes les classes DTO sont implémentées par l'interface IdentifiedDTO.

6.5.2.4.2.1.2 IdentifiedDTO

IdentifiedDTO est une interface qui reçoit un élément de type ID et contient une méthode getId() qui renvoie un élément de type ID.

Ce qui permet de connaître directement l'ID de l'objet envoyé ou reçu chaque fois qu'une classe DTO, implémentée par l'interface IdentifiedDTO, est initialisée.

6.5.2.4.2.2 Dossier Mapper

(Fichier = nom de la classe concernée plus l'annotation Mapper= NomdelatableMapper.java).

Le dossier Mapper contient vingt-trois classes, qui ont pour but la transformation d'objet de type DTO vers un objet de type Entity et Entity vers DTO.

Ces classes sont implémentées par l'interface Mapper qui prend en argument deux objets et contient deux méthodes.

Le premier renvoie un DTO et l'autre renvoie un Entity.

Ces classes ont l'annotation *@Component* (elles sont considérées comme des candidats pour la détection automatique, lors de l'utilisation de la configuration, basée sur des annotations et de l'analyse du chemin de classe), ainsi que l'annotation *@Override* qui permet l'implémentation des deux méthodes contenues dans l'interface Mapper.

6.5.2.4.2.3 Dossier Service

(Fichier = nom de la classe concernée plus l'annotation Service= NomdelatableService.java).

Ce dossier contient vingt classes, portant le nom de classe (table) associé, ainsi qu'un fichier interface.

Ces fichiers ont pour seul but la création de toutes les réponses aux questions autorisées, par le biais de l'incrémentation des méthodes de l'interface « CrudService » ou de la création de nouvelles méthodes.

Toutes les classes services ont une annotation *@Service* et contiennent les annotations *@Autowired*, *@Override* et s'il y a un ou plusieurs nouvelles méthodes, *@Transactional*.

6.5.2.4.2.3.1 @Service

Le service est une annotation de JPA indiquant qu'une classe est une « façade de service métier » (dans le sens des modèles J2EE de base), ou quelque chose de similaire.

Cette annotation est un stéréotype à usage général et les équipes individuelles peuvent affiner leur sémantique et les utiliser le cas échéant.

6.5.2.4.2.3.2 @Autowired

@Autowired est une annotation de JPA qui initialise une variable objet en donnant accès à ses méthodes publiques.

6.5.2.4.2.3.3 @Override

@Override est une annotation de JPA qui indique que la méthode est issue de l'interface implémentée et respecte la structure initiale.

- Le fichier interface :

Pour ne pas réécrire, pour chaque classe service, les méthodes CRUD (Creat, ReadOne/ReadAll, Update et Delete), je crée une interface qui va me prémâcher la création de ces cinq méthodes.

Cette interface prend en argument un objet de type DTO et un objet de Type ID.

- Les fichiers service :

Chacune des classes de type service est implémentée par l'interface « CrudService », et a besoin d'au minimum deux variables :

L'une de type Mapper et l'autre de type Repository.

Vous pouvez constater que certaines de ces méthodes procèdent selon des conditions.

Exemple pour la méthode Create :

```
if (provinceRepository.existsById(toCreat.getId()))  
    throw new ProvinceExisteException(toCreat.getId());
```

Ce qui veut dire en français :

Regarde l'identifiant de l'élément à créer et vérifie si un autre élément déjà créé possède le même identifiant.

Si oui, alors arrête le processus et crée-moi une nouvelle exception de type

`ProvinceExisteException` et donne-lui l'identifiant suivant (`toCreat.getId()`).

En effet, sachant que chaque objet créé à un identifiant unique, il ne peut donc pas avoir deux objets du même type avec le même identifiant. Il est donc préférable d'intercepter l'erreur avant.

Ce qui revient à faire :

```
try{  
    ....  
}catch (ProvinceExisteException p){  
    p(toCreat.getId())  
}
```

Autre exemple pour la modification ou la suppression :

```
if(!provinceRepository.existsById( toUpdate.getId() ))  
    throw new ProvinceFoundException(toUpdate.getId());
```

Ce qui veut dire en français :

Regarde l'identifiant de l'élément à modifier ou à supprimer et vérifie si un autre élément possède le même identifiant.

Si aucun élément déjà créé ne possède cet identifiant, alors arrête le processus et crée-moi une exception de type `ProvinceFoundException` et donne-lui l'identifiant suivant (`toUpdate.getId()`).

En effet, sachant qu'un objet créé a un identifiant unique, il nous est impossible de modifier ou supprimer un objet qui n'existe pas.

L'explication de la propriété de la classe `ProvinceExisteException` et `ProvinceFoundException` nous amène au chapitre suivant : les exceptions.

6.5.2.4.3 *Le dossier exception*

(Fichier = nom de la classe concernée plus l'annotation `ExisteException` ou `FoundException` = `NomdeLaTableExisteException/FoundException.java`).

Ce dossier contient quarante-quatre classes, portant le nom de classe (table) associé, qui sont des extensions des classes `ElementAlreadyExistsException` et `ElementFoundException`.

Ces classes ont pour seul but la capture et la personnalisation des exceptions.

6.5.2.4.3.1 *La classe `ElementAlreadyExistsException` :*

Cette abstract classe est l'extension de la classe `Exception()` et a pour but la personnalisation de cette dernière.

Elle possède une fonction qui prend en argument un objet non défini, une classe et renvoi un élément de type texte (String).

6.5.2.4.3.2 La classe *ElementFoundException* :

Cette abstract classe est l'extension de la classe `Exception()` et a pour but la personnalisation de cette dernière.

Elle possède une fonction qui prend en argument un objet non défini et une classe non définie, et renvoi un élément de type texte (String).

Une fois mise en place, ces deux classes permettent la personnalisation d'exception pour n'importe quelle classe.

La création d'un nouvel objet de type Province est un processus qui peut être problématique, il est donc nécessaire d'intercepter le ou les problèmes.

La création de la classe `ProvinceExisteException` est une personnalisation de ces interceptions.

Cette classe `ProvinceExisteException` a pour extension la `ElementAlreadyExistsException` et prend en argument un objet de type Integer et renvoie, en plus de cet argument, la classe ProvinceDTO.

La classe `ProvinceFoundException` suivra le même principe et sera utilisée lors de l'update et du delete.

Pour les méthodes `ReadOne` ou `ReadAll`, il n'est pas nécessaire de mettre en place une interception d'exception, car le processus de ces deux méthodes ne menace pas l'intégrité du programme. Au pire, ces méthodes peuvent renvoyer un objet vide ou une liste d'objets vide.

6.5.2.5 Les dossiers Rest-API

(Fichier = nom de la classe concernée plus l'annotation Controller = NomdeTableController.java).

Le dossier Rest-API contient seize fichiers qui représentent les portes d'entrées et sorties de toutes les classes (entité/table) autorisées, par les requêtes HTTP.

Elle contient aussi un fichier de type interface nommé CrudController, ainsi qu'un fichier nommé Abstract CrudController.

Lors de la création d'un Rest-API, nous connaissons déjà le type de requête HTTP qu'il est possible de recevoir, à savoir : Post, Get, Put et Delete.

Donc, pour chacune de nos classes (entités), il faudra créer une méthode répondant à chaque type de requête.

Dans le but d'une généralisation de méthode et pour éviter un maximum de redondances du code, j'ai opté pour une méthode d'héritage.

La création d'une interface (CrudController)

Cette interface demande un objet de type DTO qui a pour extension l'interface IdentifiedDTO, ainsi qu'un objet de type ID.

Elle contient cinq méthodes qui réinterprètent les méthodes de l'interface IdentifiedDTO (page 55).

La création d'une classe abstract (AbstractCrudController)

Cette classe demande un objet de type DTO qui a pour extension l'interface IdentifiedDTO, et est implémenté par l'interface CrudController.

Elle implémente les cinq méthodes du CrudController pour chaque chemin emprunté par les requêtes http et redirige cette requête vers le service approprié.

Cette interface et cette classe abstract permettent, aux classes héritières de traiter les requêtes de base (Post, Get, Put, Delete) et ainsi éviter les redondances.

Exemple :

On fournit à la classe AbstractCrudController deux objets (la classe *ProvinceDTO*, et un Integer) pour ensuite créer un constructeur qui prend en argument un objet de type CrudService, et renvoyer ce service.

C'est un peu comme si on lui disait :

Il y a dans ce programme un objet avec l'annotation service, cet objet prend en argument un objet de type *ProvinceDTO* et un objet de type Integer.

Si tu le trouves, utilise-le pour répondre à la requête.

6.6 Client Web

6.6.1 Angular



FIGURE 37 IMAGE ISSU DE GOOGLE - ANGULAR

Pour la réalisation du frontend (client Web), j'ai opté pour le Framework open source basé sur du Type Script (une dérivation de JavaScript) qui est Angular.

Sa logique de séparation par modules (Component) permet une réalisation d'application dynamique et très performante.

Angular me permet de suivre la logique de POO (Programmation Orienté Objet) mise en place côté backend, ainsi que la possibilité d'ajouter en son sein plusieurs librairies pour venir appuyer le code.

6.6.1.1 Les librairies

1. Bootstrap : cette librairie me permet d'harmoniser le style et l'affichage du code HTML.
2. Mat-Carousel : cette librairie me permet de gérer mes images.
3. App-Routing-Module : cette librairie me permet de gérer mes routes (routes de navigation) ainsi que leurs niveaux d'autorisation.

6.6.2 Structure du code

6.6.2.1 Les Classes

Les classes utilisées dans mon projet frontend représentent les types d'informations échangées entre mon client web et mon Rest-API.

Ils sont donc des répliques exactes de mes DTO côté backend.

Cette mise en place me permet d'avoir une cohérence dans les objets envoyés ou reçus lors des dialogues entre mon client web et mon Rest-API.

Toutes ces classes peuvent être retrouvées dans le dossier « App/objet.js ».

• Province	• Ville	• TypeDeBien
• TypeDeService	• Coordonnee	• Service
• Role	• Personne	• Bien
• Aladisposition	• Mdp	• ContactUser
• ContratBienMisEnLigne	• ContratReservation	• ImageProvince
• ImageVille	• ImageBien	• LikeBien
• Pays	• AdresseUser	• InfoBancaire
• ModifMDP		

6.6.2.2 Structure des « Components »

Le dynamisme et les performances d'Angular proviennent de sa capacité à déployer et à gérer des composants (components). Chaque composant est indépendant des autres et peut être appelé par d'autres composants.

Pour mon projet et sur base de la structure représentée sur le schéma ci-dessous, mises à part les branches service et assets, toutes les autres branches sont des composants, que ce soient des branches parents ou enfants.

Cette structure mise en place n'est pas un schéma de navigation, mais elle me permet de structurer mon code et de faciliter l'échange d'information entre composant parent-enfant et composant enfant-parent.

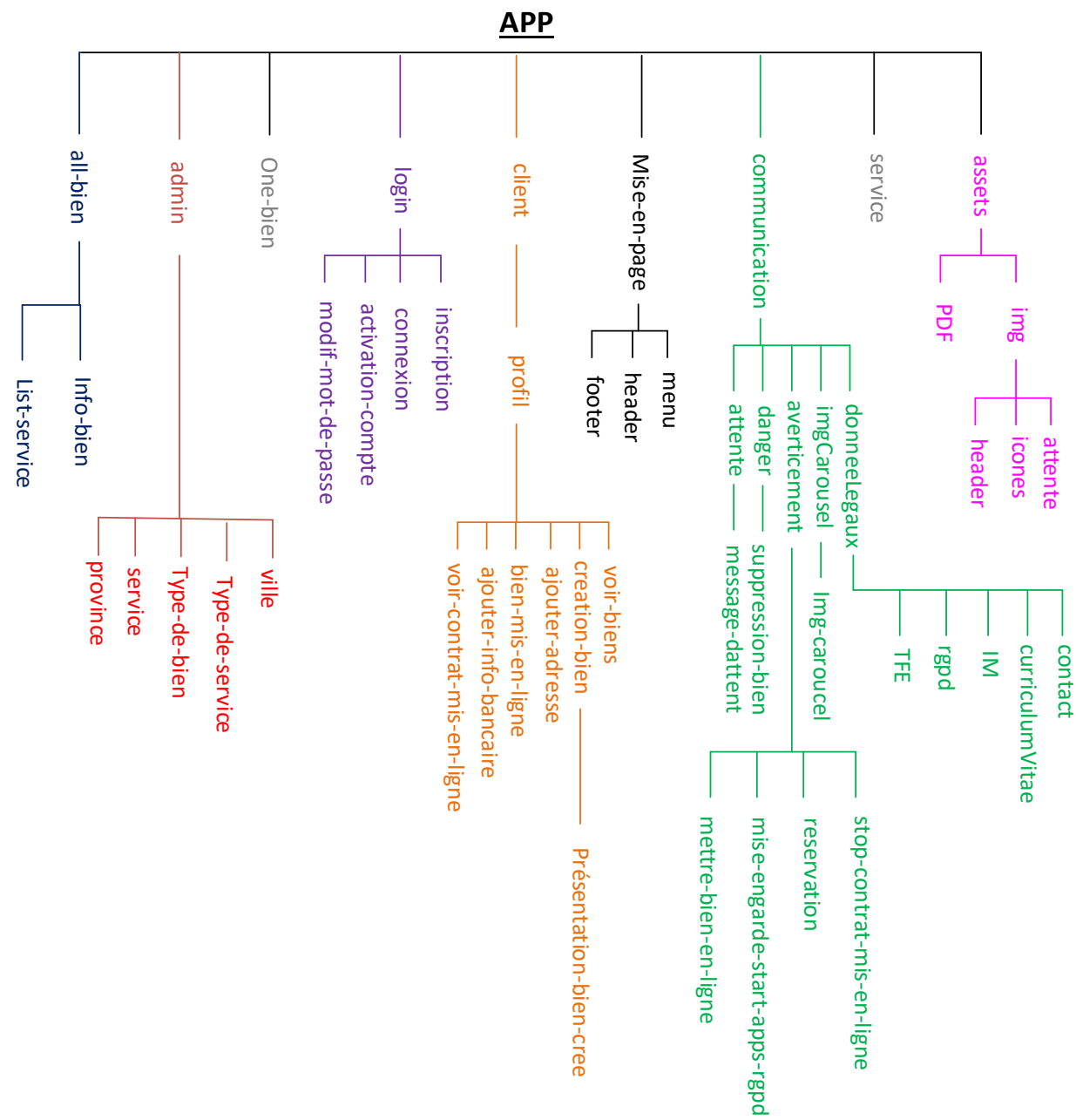


FIGURE 38 SCHEMA VISIO - STRUCTURE CLIENT ANGULAR

6.6.2.2.1 Les branches

6.6.2.2.1.1 Assets

(Ce n'est pas un component)

Le dossier assets est accessible dans tout le projet, il est destiné au stockage des fichiers image ou PDF indispensables dans la présentation et la mise en place de l'application.

Il ne contient pas d'images de présentation de biens, provinces ou villes car celles-ci sont stockées dans la base de données.

6.6.2.2.1.2 Service

(Ce n'est pas un component)

Mise à part le fichier « login.service », chaque fichier dans le dossier service contient la structure des requêtes http disponibles pour la classe visée.

Un fichier service est reconnaissable par un « nom » plus l'appellation « .service ».

Exemple :

Pour ma classe « bien » (classe qui représente les propriétés attendues lors de l'envoi ou de la réception de requête http concernant un bien), l'appellation du fichier service associé est fait comme suit => « bien.service ».

Le fichier « login.service » contient :

- Les méthodes de stockage et de récupération de variables ou objets en session Storage.
- Une méthode de redirection lors de la connexion en fonction des rôles.
- Deux méthodes de vérification des rôles.
- Une méthode de restriction de contenus Admin dans les routes URL.

Ces méthodes peuvent être appelées par les components ou les services

6.6.2.2.1.3 Communication

(C'est un component)

Ce component est constitué d'autres components destinés aux traitements et à l'affichage des pages de communication, d'avertissement ou de danger.

6.6.2.2.1.4 Mise en page

(C'est un component)

Ce component est constitué d'autres components destinés à l'affichage de partie invariable dans l'affichage d'une page :

- Header (haut de la page).
 - Menu.
- Footer (bas de la page).

6.6.2.2.1.5 Client

(C'est un component)

Ce component est constitué d'autres components destinés à regrouper toutes les actions associées au rôle client.

6.6.2.2.1.6 Login

(C'est un component)

Ce component est constitué d'autres components destinés à la gestion, la mise en forme et l'affichage lors d'une connexion, inscription, activation de compte ou d'une modification de mot de passe.

6.6.2.2.1.7 One bien

(C'est un component)

Ce component est destiné à la préparation et l'affichage d'un bien dans sa présentation simplifier.

6.6.2.2.1.8 Admin

(C'est un component)

Ce component est constitué d'autres components destinés à regrouper toutes les actions associées au rôle administrateur.

6.6.2.2.1.9 All bien

(C'est un component)

Ce component est constitué d'autres components destinés à la gestion, la mise en forme et l'affichage de biens dans leurs présentations simplifiées.

6.7 Conclusion

Lors de l'élaboration du chapitre sur le développement de mon application, j'ai pu constater que la mise en place d'un développement en POO (Programmation Orienté Objet) est très lente et fastidieuse.

Cependant elle permet une implémentation ou une modification très rapide de chaque partie individuelle, sans impacter le reste du code, ce qui conduira à des coûts de maintenance très faibles.

7 Cybersécurité

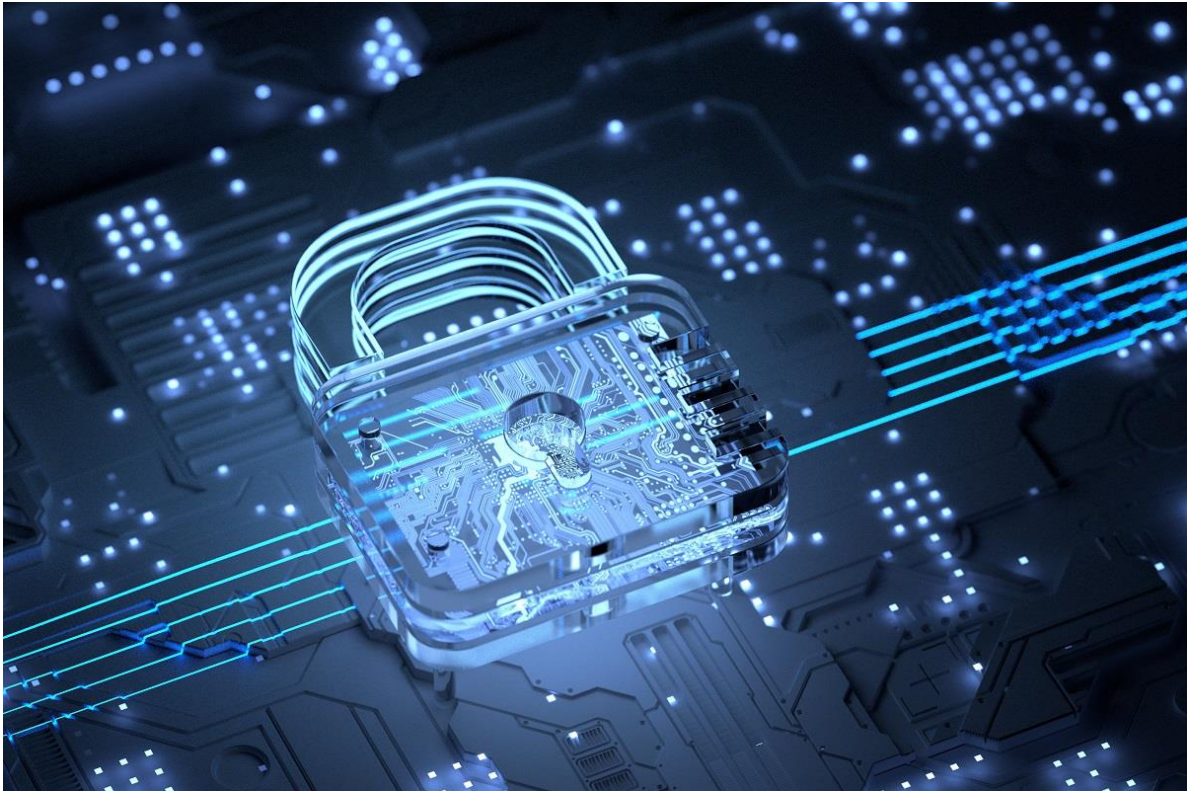


FIGURE 39 IMAGE ISSU DE GOOGLE - CYBERSECURITE

7.1 Introduction

Dans ce chapitre, je vais aborder toutes les bonnes pratiques mises en place pour préserver le système contre ma personne ainsi que contre les autres utilisateurs.

En effet, la création de l'application a levé beaucoup de questions de sécurité, qui m'ont poussé à prendre position, et à mettre en place des méthodologies ou méthodes pour faire face à ces problèmes.

7.2 Définition

Le mot cybersécurité est un néologisme désignant le rôle de l'ensemble des lois, politiques, outils, dispositifs, concepts et mécanismes de sécurité, méthodes de gestion des risques, actions, formations, bonnes pratiques et technologies qui peuvent être utilisés pour protéger les personnes et les actifs informatiques matériels et immatériels (connectés directement ou indirectement à un réseau) des États et des organisations (avec un objectif de disponibilité, intégrité et authenticité, confidentialité, preuve et non-répudiation).¹¹ (11)

¹¹ contributeur, Wikipédia. Cybersécurité. *Wikipédia*. [En ligne] L'encyclopédie libre, 2021. <https://fr.wikipedia.org/wiki/Cybers%C3%A9curit%C3%A9>.

Quelques exemples :

1. Qui a accès au contenu de l'application.
 - a. Quelle Personne ?
 - b. Quel Système ?
2. Comment garantir la clarté de mon code et garantir une compréhension rapide de celui-ci ?
3. Comment éviter qu'une personne puisse effectuer des traitements ou accéder à des informations qui ne la concernent pas ?
4. Comment protéger les informations de tous les utilisateurs ?

7.3 Protéger le système

7.3.1 Avec POO

La première étape est de faire du POO (Programmation Orienté Objet).

La mise en place d'objets, permet d'éviter la redondance et rend le code plus simple à comprendre. Il est plus facile de travailler sur quelques lignes de code que sur l'entièreté de celui-ci.

Ceci implique :

- Une maintenabilité plus rapide et efficace. En effet, lors d'une mise à jour, il sera beaucoup plus simple et rapide de trouver et de modifier un objet que de chercher dans tout un code.
- Dans un groupe, se rendre dispensable est indispensable. En effet, avoir un code bien structuré permet de rendre la compréhension et une intervention extérieure plus implicites.
- De se protéger contre nous-même et protéger notre code contre les autres. La mise en place du POO, permet de suivre une charte de construction tout au long de projet, ce qui évite notamment les fautes de frappe ou l'utilisation d'objets en dehors du contexte.
- D'alléger le contenu d'une page.

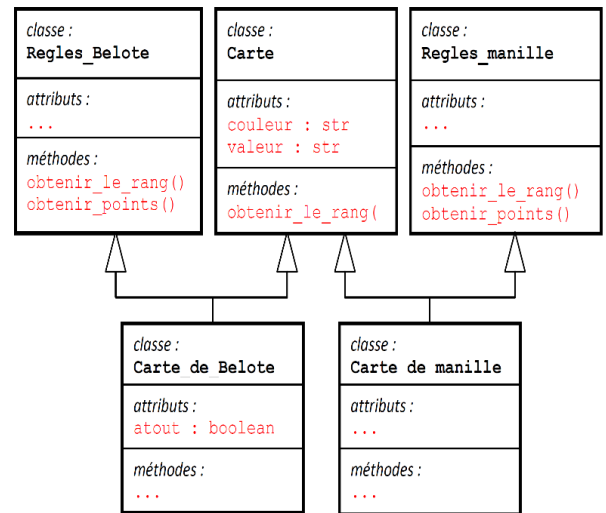


FIGURE 40 IMAGE ISSU DE GOOGLE - PROGRAMMATION ORIENTE OBJET

7.3.2 Contre les utilisateurs

L'application Mobembo.cd a été créée pour répondre à un besoin.

Pour atteindre cet objectif, il a fallu créer un cadre d'utilisation. Ce cadre permet le bon fonctionnement du système et garantit la même expérience à tous les utilisateurs.

Comment ?

- Imposer des directives d'utilisation et vérifier que ces dernières sont respectées. En effet, ce n'est pas à l'utilisateur de choisir ce qu'il veut faire, mais au système d'imposer ce qu'il peut faire.

Pour connaître les autres contraintes, veuillez retourner dans la partie « Cahier des charges » (point 4) et « Analyse » (point 5).

7.3.3 Contrôler les utilisateurs et protéger leurs données

Dans mon application, le Rest-API est chargé du contrôle et de la protection des données.

En effet, même si la nature d'une API est de répondre à une requête et aussitôt d'oublier la personne qui l'a émise, certaines requêtes nécessitent une vérification plus approfondie.

Connaitre qui est qui, ou qui peut faire quoi, est essentiel dans le système mis en place.

7.3.3.1 JWT

Pour répondre à la problématique de qui est qui, j'ai choisi la distribution de clés (token) aux clients par le billet d'un JWT (Json Web Token).

Définition :

JSON Web Token (JWT) est un standard ouvert défini dans la RFC 7519¹. Il permet l'échange sécurisé de jetons (tokens) entre plusieurs parties. Cette sécurité de l'échange se traduit par la vérification de l'intégrité et de l'*authenticité* des données. Elle s'effectue par l'algorithme HMAC ou RSA.¹² (12)

Fonctionnement :

Lorsqu'un utilisateur demande à se connecter avec des données d'identification correctes, le système lui répond en lui donnant une clé (un token). Ce token sera demandé au client à chaque requête. Ceci permettra au système de reconnaître son interlocuteur et de lui fournir les informations dont il a besoin.

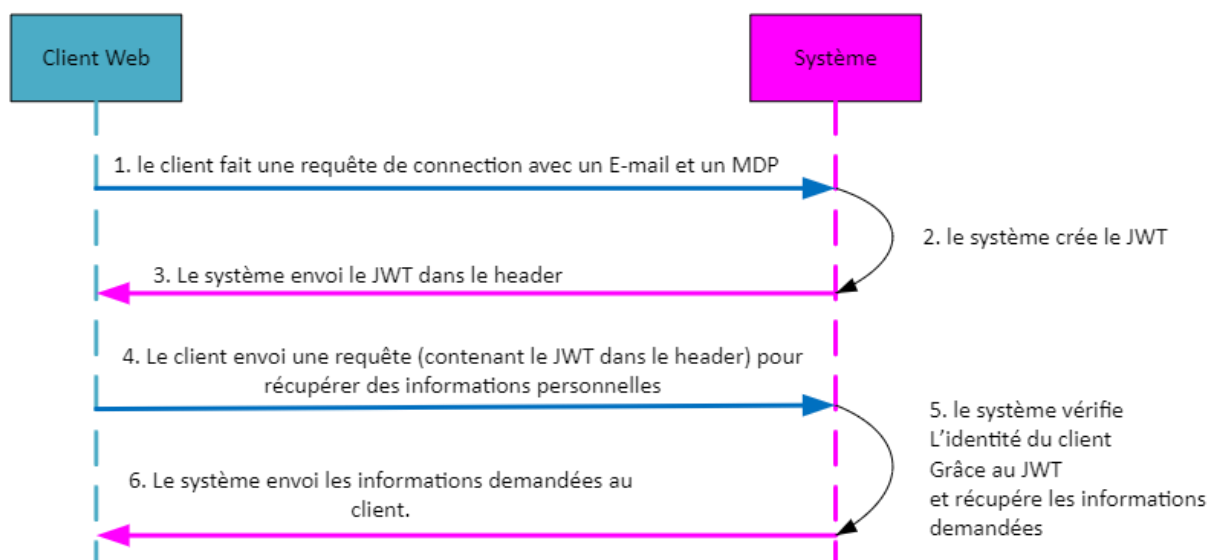


FIGURE 41 SCHEMA VISIO - GESTION DES REQUETES AVEC JWT

¹² wikipedia, Contributeur. Json Web Token. [wikipedia.org](https://fr.wikipedia.org/wiki/JSON_Web_Token). [En ligne] 2021. https://fr.wikipedia.org/wiki/JSON_Web_Token.

8 Conclusion finale

Lors de l'élaboration de mon travail de fin d'étude, j'ai pu mettre en pratique ce que j'ai acquis durant ces quatre années de bachelier en informatique de gestion.

L'un des premiers cours que j'ai eu fût d'algorithmie. Ce cours m'a suivi tout au long de mon cursus. Il m'a permis de comprendre et d'appliquer la logique métier : qui fait quoi, quand, où, mais aussi d'avoir une idée claire de chaque étape, ainsi que de chaque action à réaliser dans ce projet.

Certaines étapes ont été plus longues et plus difficiles à concevoir que d'autres. C'est pourquoi j'ai pensé judicieux de réaliser deux formations en plus de mon bachelier : JAVA Angular et .Net Cybersécurité. Ces formations m'ont aidées à accroître mes connaissances et à aller plus en profondeur pour la réalisation de mon application.

Toutefois, j'aurais aimé lors de mon analyse, avoir beaucoup plus de retour quant à la réalité de la situation en République Démocratique du Congo, tant au niveau administratif, qu'en matière des structures existantes, ceci incluant les lois nationales.

Ceci m'aurait permis la réalisation de contrats cohérents et correspondants à la réalité du terrain.

Selon moi, l'application *Mobembo.cd* n'est pas terminée. En effet, elle ne respecte pas l'un des textes du cahier des charges, qui est de mettre en ligne un bien en location. Un bien en location comprend des explications ainsi que des photos du bien en question. A l'heure actuelle, cette application est incapable de déterminer si une photo attribuée à un bien fait partie de la catégorie maison, appartement, terrain...

Pour pallier ce problème, j'ai pensé à deux solutions :

- La première solution est la création d'une IA, Intelligence Artificielle, spécialisée dans la reconnaissance d'image. Cette idée a été très vite abandonnée car elle est lourde à mettre en place et le temps imparti m'était malheureusement insuffisant.
- La deuxième solution est plus rapide à mettre en place et demande un investissement financier conséquent. Elle consiste en l'utilisation de services déjà existants, tels que ceux proposés par Amazon, Amazon Rekognition, découverts lors de mon cours d'éveil technologique.

Actuellement, une surveillance humaine est nécessaire pour le contrôle d'image de chaque bien mis en ligne, mais l'ajout d'une IA spécialisée dans l'analyse et la reconnaissance d'images permettrait d'assurer l'autonomie voulue lors de la conception de cette application.

Je reste bien conscient que lors du déploiement de cette application, une logistique conséquente doit être mise en place, comme par exemple, la création de plusieurs bureaux en RDC, ce qui permettrait aux propriétaires de laisser leurs clés ou aux locataires d'avoir un lieu de référence.

La conception de Mobembo m'a pris beaucoup de temps, je me suis énormément investi et grâce à cet accomplissement je me suis rendu compte que le choix de reprendre des études en informatique de gestion, un domaine qui m'est cher, fut une bonne décision.

Malgré mon envie d'aider la population de la RDC, une question m'a interpellé :

Ma perception de leur réalité n'est-elle pas biaisée, si non, pourquoi mes aînés utilisent le terme :
« je vais me ressourcer au pays... » ?

Bibliographie

1. Gaudiaut, Tristan. Où le tourisme est un moteur de l'économie. *statista.com*. [En ligne] 29 avril 2020. <https://fr.statista.com/infographie/21543/contribution-du-tourisme-dans-le-pib-des-pays/>.
2. Bompengo, Jahn. RDC: La contribution du tourisme au PIB est faible. *radiookapi.net*. [En ligne] 27 Septembre 2016. <https://www.radiookapi.net/2016/09/27/actualite/economie/rdc-la-contribution-du-tourisme-au-pib-est-faible#:~:text=Le%20tourisme%20a%20contribu%C3%A9%20%C3%A0,%2Dt%2Di%20fait%20savoir..>
3. Salaire minimum - Congo_kinshasa. *votresalaire.org*. [En ligne] 1 juillet 2019. <https://votresalaire.org/congo/salaire/salaire-minimum/archive/20190701>.
4. Ouest-France. Vacances d'été. Avec 852 €, le budget moyen des Français en baisse. *ouest-france.fr*. [En ligne] 3 juillet 2019. <https://www.ouest-france.fr/economie/tourisme/vacances-d-ete-avec-852-eu-le-budget-moyen-des-francais-en-baisse-6428489#:~:text=Vacances%20d'%C3%A9t%C3%A9,-,Avec%20852%20%E2%82%AC%2C%20le%20budget%20moyen%20des%20Fran%C3%A7ais%20en%20baisse,sur%20leur%20c>.
5. Caroline. Guide touristique pour visiter la Rd Congo et bien préparer son voyage. *alibabuy.com*. [En ligne] https://www.alibabuy.com/guide-touristique/rd_congo.html.
6. JetBrains. IntelliJ IDEA. *jetbrains.com*. [En ligne] 2022. <https://www.jetbrains.com/idea/promo/>.
7. Bourdon, Romain. WampServer. *wampserver.com*. [En ligne] <https://www.wampserver.com/>.
8. Oracle. MySQL. *mysql.com*. [En ligne] 2022. <https://www.mysql.com/fr/>.
9. —. Java. *dev.java*. [En ligne] 2021. <https://dev.java/>.
10. Gougler. Angular. *angular.io*. [En ligne] 2010-2022. <https://angular.io/>.
11. contributeur, Wikipédia. Cybersécurité. *Wikipédia*. [En ligne] L'encyclopédie libre, 2021. <https://fr.wikipedia.org/wiki/Cybers%C3%A9curit%C3%A9>.
12. wikipedia, Contributeur. Json Web Token. *wikipedia.org*. [En ligne] 2021. https://fr.wikipedia.org/wiki/JSON_Web_Token.