



REBUPLIQUE DU BENIN

— • — • —



MINISTERE DE L'ENSEIGNEMENT SUPERIEUR DE LA RECHERCHE
SCIENTIFIQUE (MESRS)

— • — • —

UNIVERSITE PROTESTANTE DE L'AFRIQUE DE L'OUEST (UPAO)

— • — • —

INSTITUT DES SCIENCES DE GESTION ET DE MANAGEMENT (ISGM)

— • — • —

MEMOIRE DE FIN DE CYCLE POUR L'OBTENTION DE LICENCE
PROFESSIONNELLE

— • — • —

FILIERE : Système Informatique et Logiciel (SIL)

OPTION : Sciences Techniques

THEME

**Conception et réalisation d'un système de contrôle
d'Identification : Cas de la Police**

Réalisé par :

HOTEGNI Gbètogo Serge Eric

Sous la direction du :

Maître de stage & encadreur de mémoire :

M. Aimé Gildas Yémalin AHISSOU

Ingénieur de conception en
Génie Informatique et Télécommunications

ANNEE UNIVERSITAIRE : 2020 – 2021

AVERTISSEMENT

L'Université Protestante de l'Afrique de l'Ouest (UPAO) n'entend donner ni approbation ni improbation aux opinions émises dans ce mémoire. Ces opinions doivent être considérées comme propres à leur (s) auteur (s).

DEDICACE

Ce travail est dédié à:

- Mon Père Hotegni Z. Pierre
- Ma Mère TCHOUKPA A. Euphrasie
- Mes Sœurs et frères

...pour tout le sacrifice, l'amour et la confiance que vous avez toujours manifestés à mon égard.

REMERCIEMENTS

Le présent mémoire n'a pu se réaliser sans l'assistance de l'esprit du Divin, la participation et l'appui de plusieurs personnes à qui j'exprime mes profondes gratitude. Je tiens sincèrement à remercier tous ceux qui, de près ou de loin, m'ont apporté leur assistance technique, financière ou morale pour l'aboutissement de ce projet.

Je ne peux finir sans remercier particulièrement :

- ✚ Monsieur AHISSOU Y. Gildas Aimé Mon encadreur de mémoire et maître de stage qui n'a ménagé aucun effort pour m'accorder chaque fois son temps, une oreille attentive à mes inquiétudes et des conseils avisés pour me guider ;
- ✚ Tous les contributeurs qui ont préféré rester anonymes pour leur confiance, leurs conseils et toute la documentation utile fournie.
- ✚ Mes enseignants qui ont fait de moi ce que je suis actuellement;
- ✚ Toute l'administration de l'université Protestante de l'Afrique de l'Ouest (UPAO) pour leur soutien ;
- ✚ Les honorables membres du jury, c'est un hommage que vous nous faites en acceptant de juger ce travail. Nous restons persuadés que vos critiques et suggestions contribueront à améliorer la qualité de ce travail ;
- ✚ Toute ma famille

A tous mes camarades de promotion, pour les joies et les peines que nous avons vécues ensemble ; du courage pour la suite.

LISTE DES SIGLES ET ACRONYMES

ACID :	Atomicité Consistance Isolation Durabilité
API:	Application Programming Interface
ASP .NET:	Active Server Pages for dotNet
ASP:	Active Server Pages
CGI:	Common Gateway Interface
CLR:	Common Language Runtime
CoreCLR:	Core Common Language Runtime
CSS:	Cascade Style Sheet
ECMA :	European Computer Manufacturers Association
FTP:	File Transfert Protocol
GDI :	Graphic Device Interface
HMAC :	Hash-Based Message Authentication Code
HTML:	HyperText Markup Language
HTTP:	HypertText Transfert Protocol
HTTPS :	HypertText Transfert Protocol Secure
IDE:	Integrated Developpement Environment
IETF :	Internet Engineering Task Force
IIS:	Internet Information Service
ISAPI :	Internet Server Application Program Interface
JIT:	Just In Time
JSON:	JavaScript Object Notation
JVM :	Java Virtual Machine
JWT:	JSON Web Token
LINQ :	Language INtegrated Query
MERISE:	Méthode d'Etude et de Réalisation des Systèmes d'Entreprise
MVC:	Model-View-Controller

REST:	REpresentational State Transfert
RFC:	Request For Comments
RSA:	Rivest-Shamir-Adleman
SGML:	Standard Generalized Markup Language
SSH:	Secure Shell Protocol
SSL:	Secure Socket Layer
UML:	Unified Modelling Language
UWP:	Universal Windows Platform
VS :	Visual Studio
WPF:	Windows Presentation Foundation
XAML:	eXtensible Application Markup Language

LISTE DES TABLEAUX

Tableau 1: Liste des paramètres du JWT et leur signification	37
Tableau 2: Tableau de convention des valeurs des paramètres jwt utilisés	51

LISTE DES IMAGES ET DIAGRAMMES

Figure 1 : Organigramme de KONECT Technologies	5
--	---

Image 1: Interface de Connexion / Authentification	42
Image 2: Interface du Centre d'Opérations (Vue Développeur, Utilisateur Invalide) ..	43
Image 3: Interface de vue sur les détails d'une personne physique	44
Image 4: Interface de vue sur la Gestion des Agents.....	45
Image 5: Interface de vue sur la Recherche ou l'identification d'une Personne	46

Tableau 1: Liste des paramètres du JWT et leur signification.....	37
Tableau 2: Tableau de convention des valeurs des paramètres jwt utilisés.....	51

RESUME DE LA RECHERCHE

Toute nation a besoin de veiller sur son peuple. Pour ce faire, la mise en place d'un système d'identification mobile s'avère être une des meilleures solutions; ceci pouvant permettre aux forces de l'ordre de rechercher de se renseigner sur n'importe quel individu dans le cadre d'une enquête ou une constatation. C'est d'ailleurs le cas de notre milieu d'étude où il manque un canal moderne dédié à la diffusion de ces informations-là. Notre projet consiste donc à apporter une proposition de solution à ce problème. Pour cela nos recherches approfondies sur les différents acteurs intervenants et des besoins actuels nous a mené à la conception de ce système qui a nécessité une méthode d'analyse spécifique telle que UML; La réalisation quant à elle a employé des outils de développement bien adaptés au contexte de l'application.

Mots clés : Unified Modeling Language, Système, Base de Données, API, Android, Outils de Développement...

ABSTRACT

Nowadays, Every nation needs to protect it people. Setting up an identification system seems to be an ideal way to reach that goal; allowing police agents to seek and find informations about any individual when investigation cases. Looking at the case we are studying, we have noticed that there was a lack of a modern dedicated channel which will allow police agents to send and receive those informations. Our project is about to propose an operational solution to solve that problem.

Keywords : Dedicated Channel, Informations, Programing Langage, Operational Solution, Software, Police...

SOMMAIRE

Introduction.....	1
CHAPITRE 1 : CADRE INSTITUTIONNEL DE L'ETUDE	2
I. Présentation de la structure	3
1- Historique de KONECT Technologies.....	3
2- Missions de KONECT Technologies	3
3- Fonctionnement de la structure d'accueil.....	4
II. Nos Missions au sein de la structure	6
CHAPITRE 2 : CADRE THEORIQUE DE L'ETUDE ET CONCEPTION DU SYSTEME	9
1. Présentation du thème et Objectifs de recherche	10
1- Intérêt de l'étude et énoncé du problème et critiques de l'existant	10
2- Description des objectifs et solution proposée	11
2. Présentation et choix de l'outil d'analyse	12
1. Présentation des outils d'Analyse.....	12
2. Choix de l'Outil d'Analyse	16
3. Conception du système	16
1. Modélisation Comportementale ou Fonctionnelle	16
2. Modélisation Statique.....	22
3. Modélisation Dynamique	25
CHAPITRE 3 : REALISATION DE L'APPLICATION	30
I. Choix et présentation des outils de développement	31
II. Présentation du prototype	42
III. Mesures de sécurité de l'application	47
1. L'architecture de l'environnement d'Exécution.....	47
2. La Cryptographie.....	52
3. Secure Socket Layer (SSL) / TLS (Transport Layer Security).....	53
4. La Restriction d'adresses IP (Internet Protocol).....	54
5. Authentification par empreinte digitale	54
Conclusion	55
Références Bibliographiques.....	x
Webographie.....	xi

INTRODUCTION

La conception et la mise en place d'un système d'identification à l'échelle nationale revêt une importance capitale pour toute nation soucieuse de sa sécurité. Un système de contrôle d'identification est un système permettant de renseigner des informations d'un groupe de personnes et d'effectuer facilement une recherche de ces personnes à travers des critères qui leur sont uniques. Pour le cas étudié, on note un manque de canal plus moderne de diffusion de l'information ainsi que le contrôle d'identité qui n'est pas à sa meilleure forme.

Dans cette optique, nous nous sommes donnés comme objectif de moderniser et d'automatiser le processus d'accès aux informations par la police — et cela — en mettant en place un système de contrôle d'identification et de diffusion de l'information afin d'améliorer les moyens mis à disposition pour l'accomplissement de sa tâche. L'objectif premier visé est de contribuer à la consolidation de la sécurité de l'Etat et de ses institutions puis ensuite rendre l'identification rapide, fiable et possible en tout lieu.

Ce projet comporte trois solutions principales dont une API privée et sécurisée qui expose un certain nombre d'informations contrôlées et authentiques, Une application Bureau servant de centre d'opérations et une application mobile Android pour acheminer l'information vers les agents aux quatre coins du pays.

C'est dans le souci d'offrir un outil fiable et performant d'aide à l'identification que nous avons été accueilli au sein de la société KONECT Technologies pour un stage d'une durée de trois (03) mois (du 29 Mars 2021 au 07 juin 2021) sur le thème «Conception et réalisation d'un système de contrôle d'Identification : Cas de la Police» qui comportera deux modules, notamment le module sur micro-ordinateurs personnels et le module sur terminaux mobiles (en particulier avec Android comme système d'exploitation).

Le présent rapport est subdivisé en trois grands chapitres. Le premier chapitre sera consacré à la présentation du cadre institutionnel de l'étude, le deuxième chapitre au cadre théorique de l'étude et à conception du système et le dernier chapitre à la conception du système.

CHAPITRE 1 : CADRE INSTITUTIONNEL DE L'ETUDE

I. Présentation de la structure

1- Historique de KONECT Technologies

KONECT Technologies, enregistré au registre du commerce et du crédit Immobilier sous le numéro COTONOU N° RCCM RB/PNO/18 A 8611 avec le numéro d'Identifiant Fiscale Unique (IFU) 120170223333203, est une entreprise qui assure la prestation de services dans les domaines de l'informatique et des télécommunications. En effet, elle assure la fourniture, l'installation, la configuration et la maintenance des équipements électroniques, informatiques et téléphoniques. Elle se charge également de la fourniture et de l'installation des équipements en énergie renouvelable notamment les équipements nécessaires à l'alimentation photovoltaïque domestique. Son siège est installé dans le cinquième arrondissement de Porto-Novo, précisément Ilot 189 dans le quartier Ouando à cinq cent (500) mètres du carrefour cinquantenaires dans la ruelle après l'Agence Béninoise de la Promotion de la Famille (ABPF).

2- Missions de KONECT Technologies

L'entreprise s'est donnée comme mission de vulgariser les Technologies de l'Information et de la Communication (TIC), les outils de gestions des activités commerciales et de l'Énergie renouvelable en assurant non seulement une formation des techniciens qualifiés, mais également des services de qualité. Aussi, elle assure la conception des solutions informatiques comme la réalisation des Progiciels, des logiciels de gestion pour les entreprises et des sites web professionnels en tout genre.

Afin de remplir efficacement cette mission, le directeur de l'Entreprise KONECT Technologies s'est associé avec des techniciens dévoués à la tâche pour assurer des formations de qualité aux jeunes résidents dans la ville capitale et ses environs.

Le principal objectif de KONECT Technologies est de rapprocher les opportunités de formation et de stage des jeunes de la ville capitale et ses environs.

Pour atteindre ce but, l'Entreprise s'est donc donné des buts spécifiques à atteindre qui sont :

- Permettre aux étudiants d'avoir une formation pratique en maintenance informatique, en dimensionnement et en installation d'alimentation photovoltaïque;

- Fournir des solutions technologique et logicielle pour faciliter la gestion des entreprises;
- Offrir un bon cadre de remise à niveau aux professionnels en activité de même qu'aux assistants de direction;
- Assurer des formations pour le dimensionnement, la configuration et l'installation des réseaux informatiques et téléphoniques;
- Assurer des formations pour la conception graphique et la conception de sites internet...

3- Fonctionnement de la structure d'accueil

Le bon fonctionnement de l'entreprise est le résultat du travail synergique de plusieurs démembrements. En effet, KONECT Technologies est dirigé par une équipe organisée en hiérarchie.

❖ La Direction Générale

Organe décisionnel, la Direction générale organise et coordonne toutes les activités de l'entreprise, Elle prend les décisions en collaboration avec les autres directions et assure le suivi de leur exécution.

❖ La Direction Exécutive

Elle assure l'intérim de la Direction Générale en cas d'absence. Dans ce cas, elle remplit les mêmes Fonctions que cette dernière, mais la prise de décision finale est laissée à l'approbation de la Direction Générale.

Elle assure également les responsabilités techniques, gère le matériel, étudie des solutions techniques pour les partenaires. Elle s'occupe de l'encadrement des stagiaires et des formations des apprentis stagiaires et enfin la programmation des remises à niveau des professionnels et des fonctionnaires (recyclage).

Elle assure et est responsable de l'étude des appels d'offre afin de proposer une réponse qui est étudiée et validée en réunion de comité. Cette direction comporte trois sections à savoir: la section informatique, la section télécommunication, et la section graphisme et web design.

❖ La Section Informatique

Elle assure la proposition des solutions informatiques aux entreprises partenaires, la fourniture des équipements informatiques et leur entretien. Elle s'occupe également de la maintenance des équipements informatiques.

❖ La Section Télécommunication

Elle propose aux entreprises partenaires des solutions dans le domaine des réseaux de télécommunication, assure la fourniture des équipements leur installation, leur configuration et leur maintenance. Elle s'occupe également de la proposition des réponses aux appels d'offre.

❖ La section Graphisme et Web Design

Elle s'occupe de la conception graphique pour la visibilité des entreprises et aussi de la conception et de l'hébergement des sites web pour les entreprises partenaires. Elle assure également la production des applications web et Android pour les partenaires.

Elle s'occupe enfin de la formation en graphisme et création web pour les stagiaires.

❖ Le Secrétariat

Rattaché à la Direction Générale, il assure la production documentaire. Il reçoit et ventile les informations dans l'entreprise. Il assure également l'accueil des clients et des partenaires et gère les courriers.

❖ La direction Commerciale

Elle s'occupe de la politique de promotion de l'entreprise gère les relations de partenariat entre les partenaires et l'entreprise. Elle s'occupe également des plans de développement de l'entreprise en proposant des perspectives pour une bonne visibilité des activités de l'entreprise pouvant permettre à cette dernière d'explorer d'autres possibilités qui ne sont pas encore exploitées.

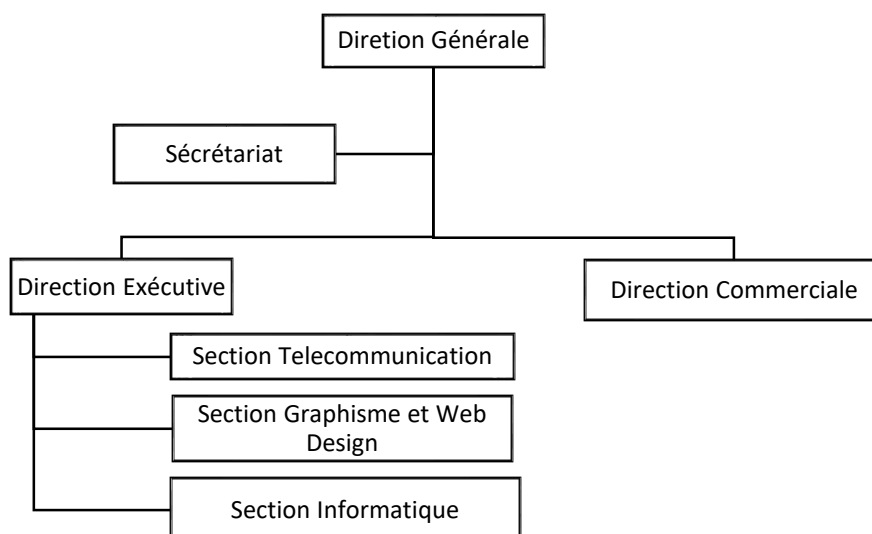


Figure 1 : Organigramme de KONECT Technologies

II. Nos Missions au sein de la structure

2.1 Déroutement du Stage

Tout le long de notre séjour au sein de l'entreprise KONECT Technologies, nous avons eu d'enrichissantes expériences en informatique, en particulier dans les domaines de la maintenance, de l'architecture des ordinateurs et du développement Web et d'Applications logicielles. Nous avons, avec attention et dévotion, appris à conduire une analyse complète dans le cadre de la conception d'une solution informatique moderne. Cela nous a amené à, notamment, renforcer nos connaissances théoriques en modélisation (UML notamment), à étudier les critères de faisabilité d'un projet informatique.

Plus excitant, nous avons eu à assister notre Maître de Stage dans la conception d'une solution logicielle baptisée *WheelTrack* de gestion pour une structure spécialisée dans la vente de produits et services dans le domaine de la pneumatique. Cela nous a amené à toucher de près les différents outils nécessaires au développement et au déploiement d'une solution Logicielle moderne dans le respect du cycle de vie de développement d'un logiciel. Nous avons par exemple appris à nous servir des outils de développements tels qu'un logiciel de gestion de version, un debugger, une source de donnée (Base de données Relationnelle et Objet), et d'un langage phare de la plateforme dotNet (Visual Basics .NET).

2.2 Déroutement de L'enquête

Pour notre sujet, en dehors de notre structure d'accueil, nous avons eu à effectuer une enquête sur le terrain renforcée par une étude documentaire afin de cerner le milieu pour lequel nous voulons développer une solution informatique.

2.2.1 La Police républicaine du Bénin

La police républicaine du Bénin dont le siège est à Cotonou est la force unique de sécurité intérieure résultant de la fusion de l'ex police nationale et l'ex gendarmerie nationale du Bénin conformément à la loi N°2017-41 du 29 décembre 2017. Elle assure le maintien de la loi au avec la douane, les Eaux et forêts, la police sanitaire, la police environnementale, la police de la pêche et l'Unité Mixte de Contrôle des Containers (UMCC).

Ses attributions principales sont d'assurer:

- La protection des institutions et installations de l'Etat;
- Le respect des lois et règlements ;
- La protection des personnes et des biens.

Il est composé de:

- Des fonctionnaires de la police nationale;
- Des militaires de la gendarmerie nationale;
- Du personnel recruté en des dispositions de statut des personnels de la Police républicaine.

La Police Républicaine du Bénin est membre de plusieurs organisations internationales de lutte contre le crime comme Interpol (depuis le 19 septembre 1962) ou encore le Programme SIPAO qui vise à répondre aux défis sécuritaires auxquels sont confrontés les pays d'Afrique de l'Ouest, en permettant une collecte efficace d'informations de police par le biais d'un système national centralisé et en offrant la possibilité de partager les informations recueillies aux niveaux national, régional et international.

Notre enquête s'effectuant dans le cadre de la mise en place d'une solution informatique, nous ne saurions continuer sans mentionner la Direction qui accueille notre enquête et nos travaux.

2.2.2 Présentation de la Direction des Systèmes d'Information et des Communications (DSIC)

La Direction des Systèmes d'Information et des Communications assure, en liaison avec toutes les structures de la Direction Générale de la Police Républicaine, la conception, la mise en œuvre et le suivi-évaluation des actions visant à permettre la fluidité et l'accessibilité de l'information et des communications de la Police républicaine.

A ce titre, elle assure :

- L'élaboration des plans de liaison radioélectriques et informatiques propres à la Police Républicaine ;
- La conception des clés de codage, de décodage des communications et des logiciels ;
- La coordination, la centralisation et le contrôle de l'ensemble des activités des réseaux, des radios et systèmes informatiques ;
- La mise en place des dispositifs de sécurité, de protection des installations et des logiciels ;
- La surveillance, la détection et la neutralisation des réseaux pirates ;
- La gestion, l'installation, la maintenance, le dépannage des matériels de transmission.

CHAPITRE 2 : CADRE THEORIQUE DE L'ETUDE ET CONCEPTION DU SYSTEME

1. Présentation du thème et Objectifs de recherche

Avec la recrudescence du terrorisme, des délits criminels et des fraudes ces dernières années, il devient important de coordonner dans la plus grande des mesures, l'action policière grâce à un système informatique robuste respectant les standards modernes de transmission et de traitement de l'information. Quel sont les intérêts de l'étude et quels sont les objectifs visés par le système que nous préconisons ?

1- Intérêt de l'étude et énoncé du problème et critiques de l'existant

L'analyse de l'existant dans le cadre de transmission et diffusion de l'information vers les agents sur le terrain a révélé que bien que le système mise en place soit éprouvé et fonctionnel, il manque un canal adéquat de diffusion de l'information qu'elle soit multimédia ou textuelle. Le cas le plus flagrant est celui où certaines informations qui ne peuvent être transmises par la manière conventionnelle Talkie-walkie (Emetteur-Récepteur) sont acheminées à destination des agents sur le terrain via leurs téléphones mobiles, plus précisément par une application tierce : WhatsApp. Ce qui peut notamment compromettre les attributions principales de la Police dans le cas d'interception de l'information.

Autre problème important qui nous a été révélé suite à notre enquête provient du fait que le contrôle d'identité est resté à sa plus basique implémentation qui est de vérifier la carte d'identité d'une personne physique. Or, nous savons que, par un mécanisme de fraude — contre lequel luttent les forces policières depuis des années — certains individus malintentionnés en viennent à s'établir plusieurs cartes d'identité fictives dans le but de passer au travers des mailles des vérifications rigoureuses. Il convient donc de disposer d'un système d'identification qui pourrait permettre aux agents sur le terrain, à n'importe quel endroit et à n'importe quel moment de vérifier les informations présentées par un potentiel fraudeur.

Dans le but de pallier à ces différentes insuffisances que nous avons détectées, nous avons décidé de mettre au profit de la DSIC un système de contrôle d'identification couplé à un système dédié et fiable de diffusion de l'information à travers le thème formulé comme suit :

**Conception et réalisation d'un système de contrôle d'identification :
Cas de la police.**

Les questions essentielles auxquelles notre étude pourrait nous amener à répondre sont les suivantes :

- Comment peut-on permettre aux agents d'accéder en tous lieux aux informations nécessaires aux fins de vérifier l'identité d'une personne physique ?
- Comment peut-on permettre à la police de renforcer ses méthodes d'enquête en ayant à sa disposition une base organisée d'informations sur toutes les personnes présentes sur le territoire national ?
- Comment diffuser de façon fiable des informations à l'endroit de tous les agents en général ou d'un groupe d'agents en particulier
- Comment propager aussi rapidement que possible un avis de recherche afin de permettre aux agents de mener les fouilles nécessaires avec une plus grande réactivité ?

2- Description des objectifs et solution proposée

A partir des insuffisances qui ont précédemment été étudiées et dans un souci de concevoir et de réaliser un système d'informations et de contrôle d'indentification, nous avons décidé de mettre en place un système construit autour d'une API (Application Programming Interface) web permettant aux agents de police d'avoir accès à une base sécurisée d'informations en continu au travers de terminaux mobiles et desktop via des applications dédiées ; Cela en correspondance avec les droits et privilèges qui régissent chacun des acteurs du système.

L'énoncé de l'objectif général vient également avec de nombreux point spécifiques comme :

- La prise en charge du processus de notification et de diffusion de l'information précieuse ;
- La mise en œuvre d'un mécanisme de sécurisation et de contrôle de l'authenticité de l'information plus adapté au contexte présent ;
- La mise en place d'un système fiable d'identification d'une personne physique
- L'amélioration de la réactivité lors des enquêtes ou des constatations policières.
- La traçabilité numérique : L'enregistrement des données d'activités dans la base de données.

2. Présentation et choix de l'outil d'analyse

1. Présentation des outils d'Analyse.

Il existe une pléthore d'outils d'analyse et de conception dont MDA, OMT, BPEL, 3AR, MERISE, UML, MASE, SART. Cependant, seuls deux outils nous ont été enseignés tout le long de notre formation. Il s'agit de MERISE et UML.

1.1. La Méthode MERISE

La Méthode d'Etude et de Réalisation Informatique par Sous-Ensembles (MERISE) est une méthode d'analyse et de conception des systèmes d'information basée sur le principe de séparation des données et des traitements. Elle possède plusieurs modèles qui sont répartis sur 3 niveaux à savoir : Le niveau Conceptuel, le niveau Logique et le niveau Physique.



Merise propose alors une démarche articulée selon trois (3) axes pour hiérarchiser les préoccupations et les questions auxquelles répondre lors de la conduite d'un projet:

➤ Le Cycle de vie réparti en :

- Phase de Conception;
- Phase de Réalisation;
- Phase de Maintenance;

➤ Le Cycle de décision qui consiste à :

- Faire une étude détaillée du projet;
- Prendre les décisions utiles à la mise en œuvre du projet;
- Documenter les étapes de mise en œuvre du projet;

➤ Le Cycle d'Abstraction divisé en :

- Niveau Conceptuel
- Niveau d'Organisation
- Niveau Logique
- Niveau Physique

MERISE représente une méthode testée, éprouvée et très utilisée par les grandes firmes informatiques dans le but de modéliser leurs systèmes informatiques. C'est une méthode qui a évolué au fil du temps en s'adaptant continuellement aux révolutions du monde informatique. Sa maîtrise est donc un atout important dans la réalisation de notre projet.

1.2. UML

UML (Unified Modeling Language) est une synthèse de langages de modélisation objet antérieurs (Booch, OMT, OOSE..) et dont la dernière version en date est la version 2.5 validée par l'OMG (Object Management Group) en 2017. UML est destiné à faciliter la conception des documents nécessaire au développement d'un Système informatique en utilisant le paradigme Orienté objet comme standard de modélisation de l'architecture logicielle. Il est décomposé en plusieurs parties à savoir:



○ Les vues

Ce sont les observables du système. Elles décrivent le système d'un point de vue donné qui peut être organisationnel, dynamique, temporel, architectural, géographique, logique, etc. En combinant toutes les vues, il est possible de retrouver le système complet. Il existe divers types de vues : Vue des cas d'utilisation, Vue logique, Vue d'implémentation, Vue des processus, et Vue de déploiement.

○ Les diagrammes

Ce sont des ensembles d'éléments graphiques qui décrivent le contenu des vues qui sont des notions abstraites et peuvent faire partie de plusieurs vues. En UML,

les diagrammes sont hiérarchiquement dépendants et se complètent de façon à permettre la modélisation d'un projet tout au long de son cycle de vie.

Il existe quatorze(14) types de diagrammes regroupés en trois (3) grandes catégories:

❖ Les Diagrammes de structure ou diagrammes statiques

- Diagramme de classes : représentant les classes intervenant dans le système
- Diagramme d'objets: représentant les instances des classes utilisées dans le système
- Diagramme de Composants: représentant les composants du système d'un point de vue physique, tels qu'ils sont mis en œuvres.
- Diagramme de déploiement: représentant les éléments matériels (ordinateurs, réseaux, stockage...) et la manière dont les composants sont répartis sur ces éléments matériels et interagissent entre eux.
- Diagramme de paquets: représentant des dépendances entre les paquets, c'est-à-dire entre les ensembles de définition (des classes).
- Diagramme de structure composite : Représentant les relations entre composants d'une classe sous la forme de boîte blanche.
- Diagramme de profils : Spécialisation et personnalisation pour un domaine particulier d'un méta-modèle de référence d'UML.

❖ Les Diagrammes de Comportement

- Diagramme des cas d'utilisation: représentant les possibilités d'interaction entre le système et les acteurs. C'est-à-dire toutes les fonctionnalités que le système doit fournir.
- Diagramme états-transitions: représentant finis les comportements du système ou de ses composants sous forme de machines à états.

- Diagramme d'activité : représentation sous forme de flux ou d'enchaînement d'activités du comportement du système ou de ses composants.

❖ **Diagrammes d'interaction ou diagrammes dynamiques**

- Diagramme de séquence : représentation de façon séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.
- Diagramme de communication : représentation de façon simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.
- Diagramme global d'interaction: représentation des enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagramme de séquence.
- Diagramme de temps : représentation des variations d'une donnée au cours du temps.

○ Les modèles d'élément :

Il s'agit tout simplement les éléments graphiques des diagrammes.

UML, malgré sa récence, a su s'imposer dans le monde informatique notamment grâce à son concept objet qui apporte un haut niveau d'abstraction et d'encapsulation, de ses diagrammes qui permettent un prototypage rapide convenablement aux langages de programmation modernes qui sont pour la plupart Orientés Objet.

Nous discuterons par la suite de notre choix quant à la méthode d'Etude qui conviendrait à la réalisation de notre projet ainsi que des raisons éventuelles qui justifient ce choix.

2. Choix de l'Outil d'Analyse

Un Système d'information moderne utilisant un langage de programmation moderne (orienté-objet notamment), et qui implique l'interaction de plusieurs acteurs ainsi que différentes contraintes de mise en œuvre vient avec une complexité qui exige une approche exhaustive et qualitative afin de prendre en compte la plupart de nos objectifs spécifiques.

L'approche par les modèles permet de mettre en lumière tous les aspects du projet qui pourraient passer inaperçus au cours de la phase de conception. La méthode que nous jugeons adaptée pour résoudre ce type de défis est sans aucun doute UML au vu de son caractère universel, de ses avantages en termes de modélisation et de son support inné pour le paradigme Orienté-Objet.

3. Conception du système

En Abordant la démarche UML, nous avons déduit quelques diagrammes essentiels à la réalisation de notre projet. Nous en avons relevés six (06) qui sont : Les diagrammes de Classe, de Déploiement, de Cas d'Utilisation, d'Activité, d'Etats-Transitions et de Séquence.

1. Modélisation Comportementale ou Fonctionnelle

1.1. Les acteurs du système

Un acteur représente une entité appartenant à l'environnement de l'application qui interagit avec l'application. Le concept d'acteur permet de classer les entités externes à l'application. Un acteur est identifié par un nom. Dans un système informatique moderne, il existe trois (03) catégories d'acteurs :

- Les acteurs Humains
- Les acteurs Logiciels
- Les Matériels

Pour notre étude, nous avons identifié trois (03) types d'acteurs qui sont :

- Agent
- Opérateur
- Agent Administrateur

Ces différents acteurs jouent et utilisent de système de diverses manières conformément à leurs privilèges. Il convient également de noter que tous ces acteurs appartiennent à la catégorie des acteurs Humains.

1.2. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est un diagramme UML utilisé pour une représentation du comportement fonctionnel d'un système logiciel. Un cas d'utilisation (Use case) représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. En ce qui concerne les relations entre les différentes composantes du diagramme de cas d'utilisation, UML admet en tout trois (03) types de relations qui sont : L'inclusion, l'Extension et enfin la généralisation (qu'il est possible d'appliquer aux acteurs).

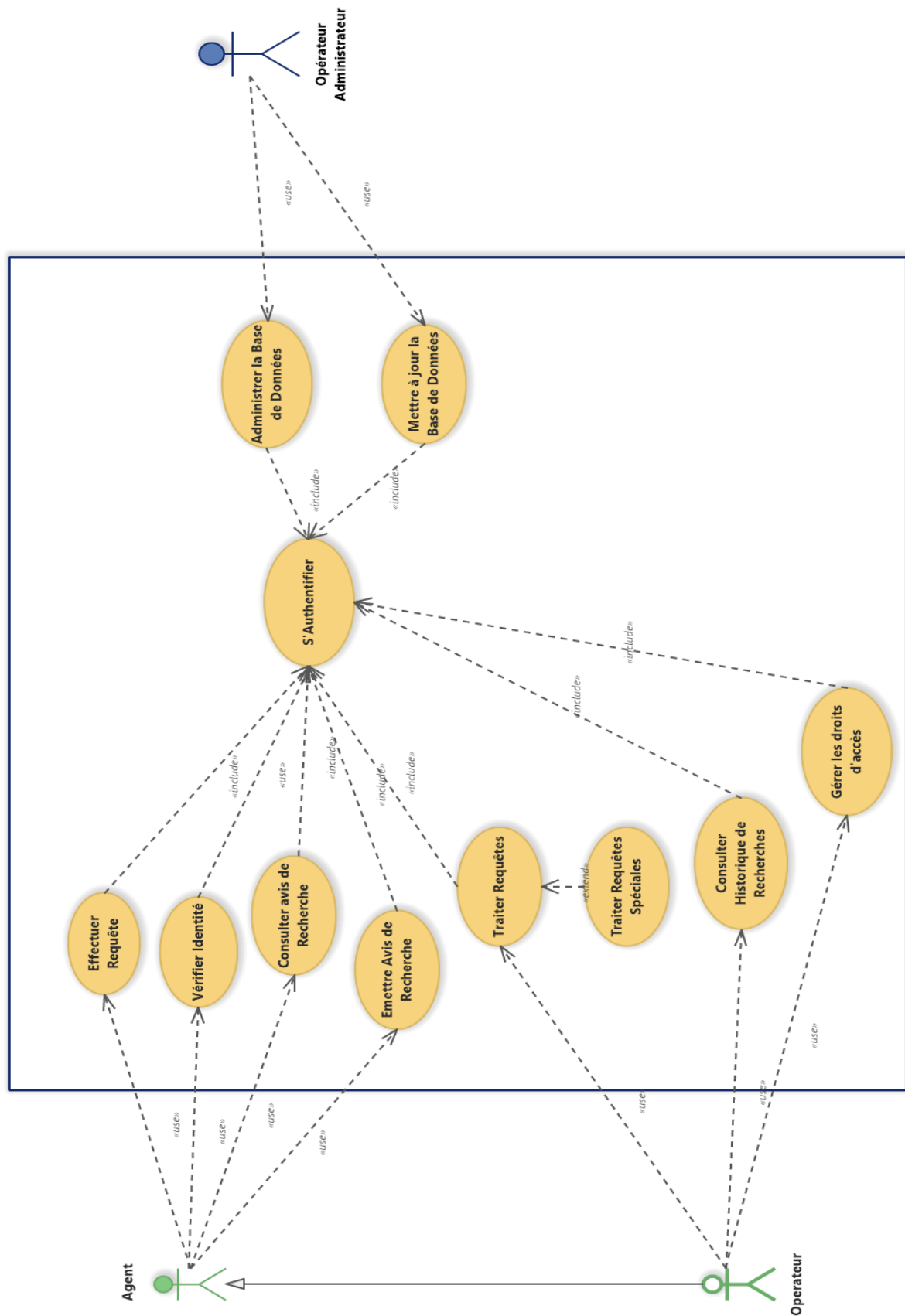


Diagramme 1: Diagramme de Cas d'Utilisation du Système

1.3. Diagramme d'Etats-Transitions

Ce diagramme est sans doute le plus important pour rendre, modéliser et comprendre les interactions entre les différentes Entités et composantes du système. Son Objectif est de décrire le comportement dynamique d'une entité (logiciel, composant, objet...). Pour cela, l'on se sert d'une description qui combine les « Etats » et les « Transitions » entre les Etats.

L'Etat étant défini comme étant l'abstraction d'un moment de la vie d'une entité pendant lequel elle satisfait à des conditions données. Les Transitions représentent alors le changement d'Etat.

Pour comprendre le mécanisme d'échange de données que nous avons mis en place pour notre projet, Nous avons décidé d'effectuer un rendu générique et exhaustif du Système par le biais des Etats-Transitions. Le diagramme ci-dessous représente ce tel rendu.

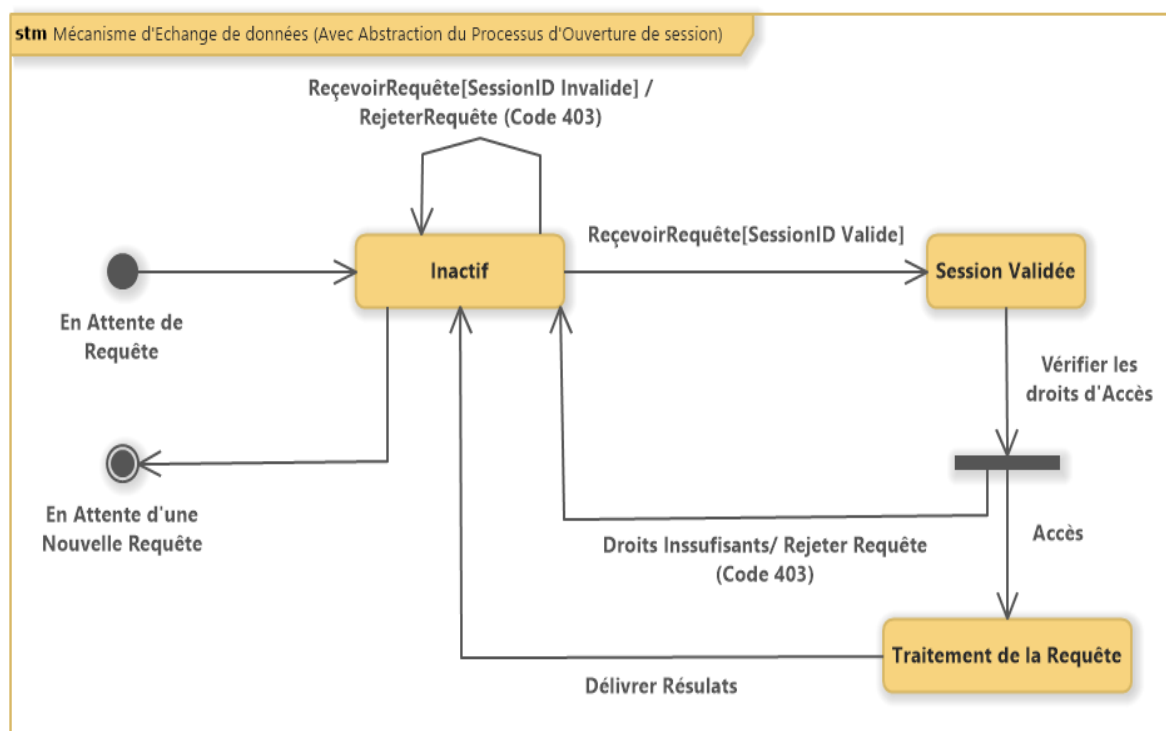


Diagramme 2: Diagramme d'Etats-Transitions du système

1.4. Diagrammes d'Activité

Le diagramme d'activités est une variante du diagramme d'états-transitions (états et transitions). Les deux types de diagrammes permettent d'avoir deux vues différentes du même système. Il visualise un graphe d'activités qui modélise le comportement interne d'une méthode (réalisation d'une opération), d'un cas d'utilisation ou plus généralement d'un processus impliquant un ou plusieurs classificateurs (classes - cas d'utilisation - paquetages etc.). Voici ci-dessous quelques diagrammes d'Activités que nous avons qu'il est utile de présenter :

- Processus d'Ouverture d'une nouvelle Session

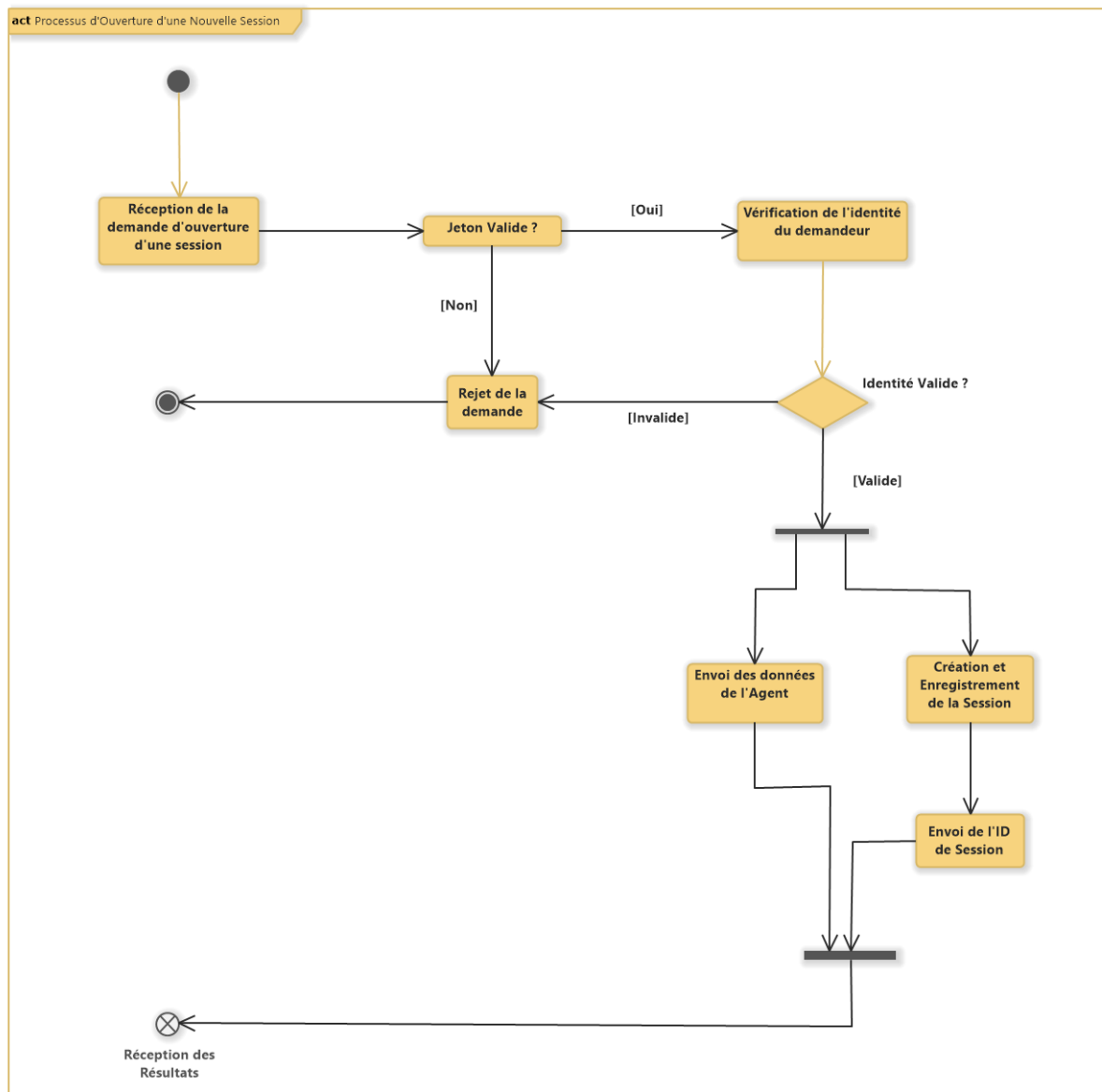


Diagramme 3: Diagramme d'Activités pour l'Ouverture d'une nouvelle Session

- Opération de Création et de diffusion d'un Nouvel Avis de recherche

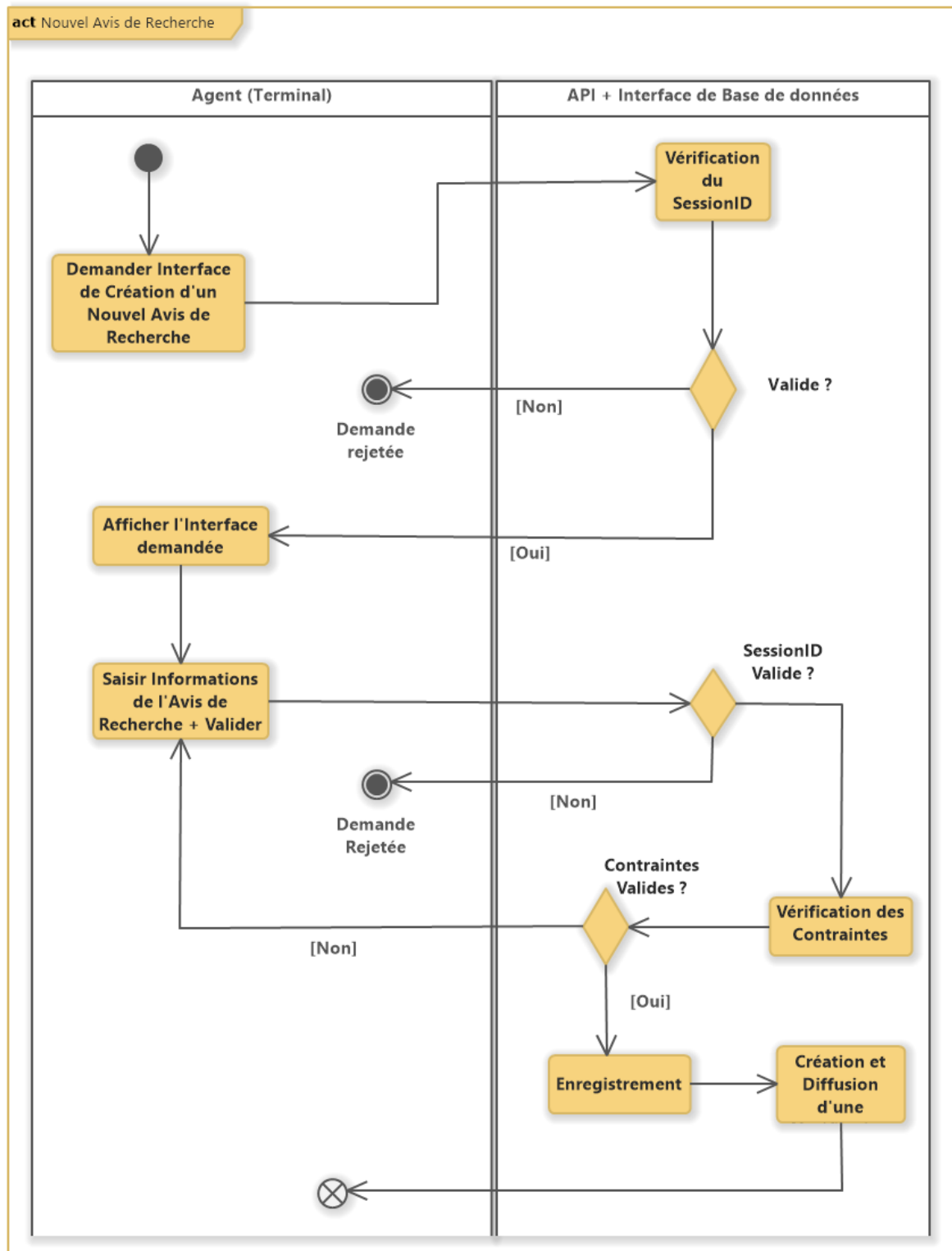


Diagramme 4: Diagramme d'Activités pour L'Emission d'un Avis de Recherche

▪ Rechercher ou Identifier une Personne

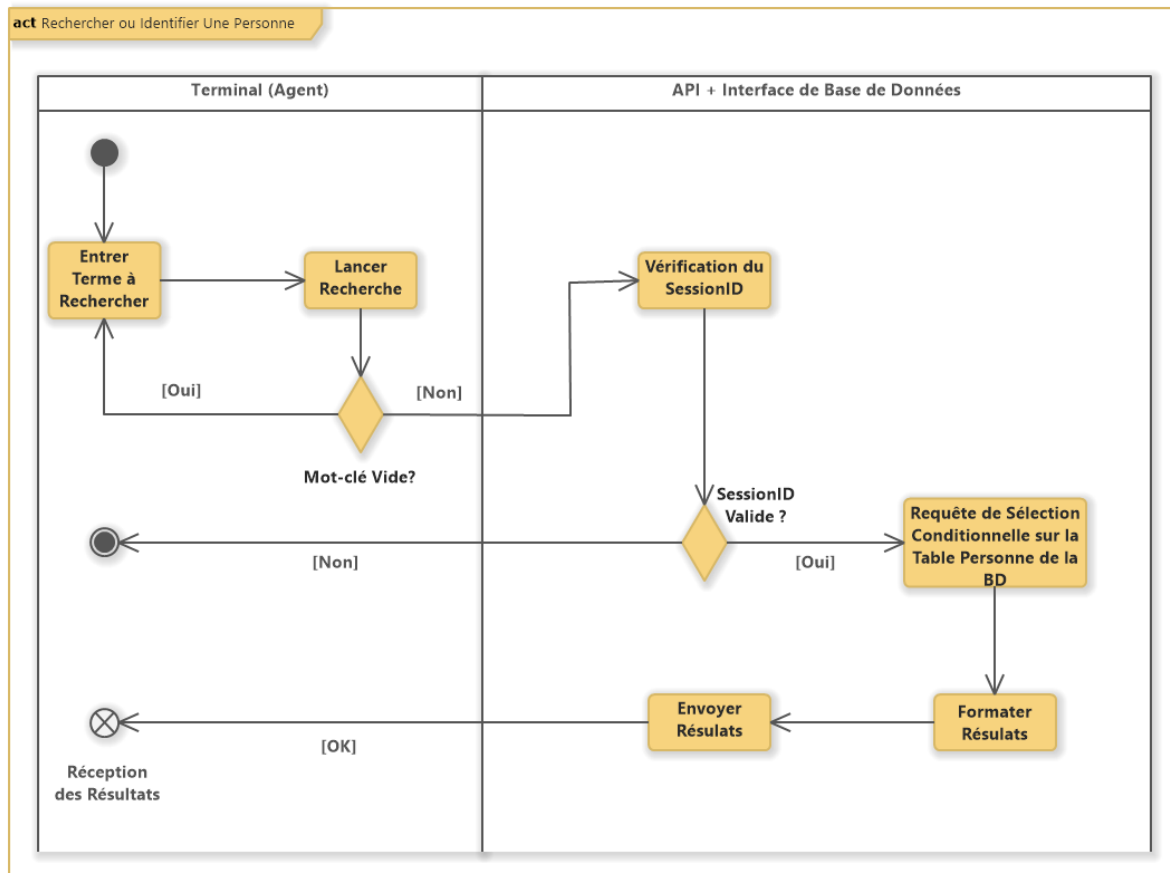


Diagramme 5: Diagramme d'Activité pour l'identification d'une personne

2. Modélisation Statique

2.3 Diagramme de classes

Les diagrammes de classes sont l'un des types de diagrammes UML les plus utiles, car ils décrivent clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets. En UML, Une classe définit la structure commune d'un ensemble d'objets et permet la construction d'objets instances de cette classe. Le concept d'association entre deux classes permet de préciser les relations qui peuvent exister entre plusieurs objets.

A l'issue de notre étude du système à concevoir, nous avons modélisé dans le diagramme ci-dessous les relations entre les différentes classes que nous avons identifiées.

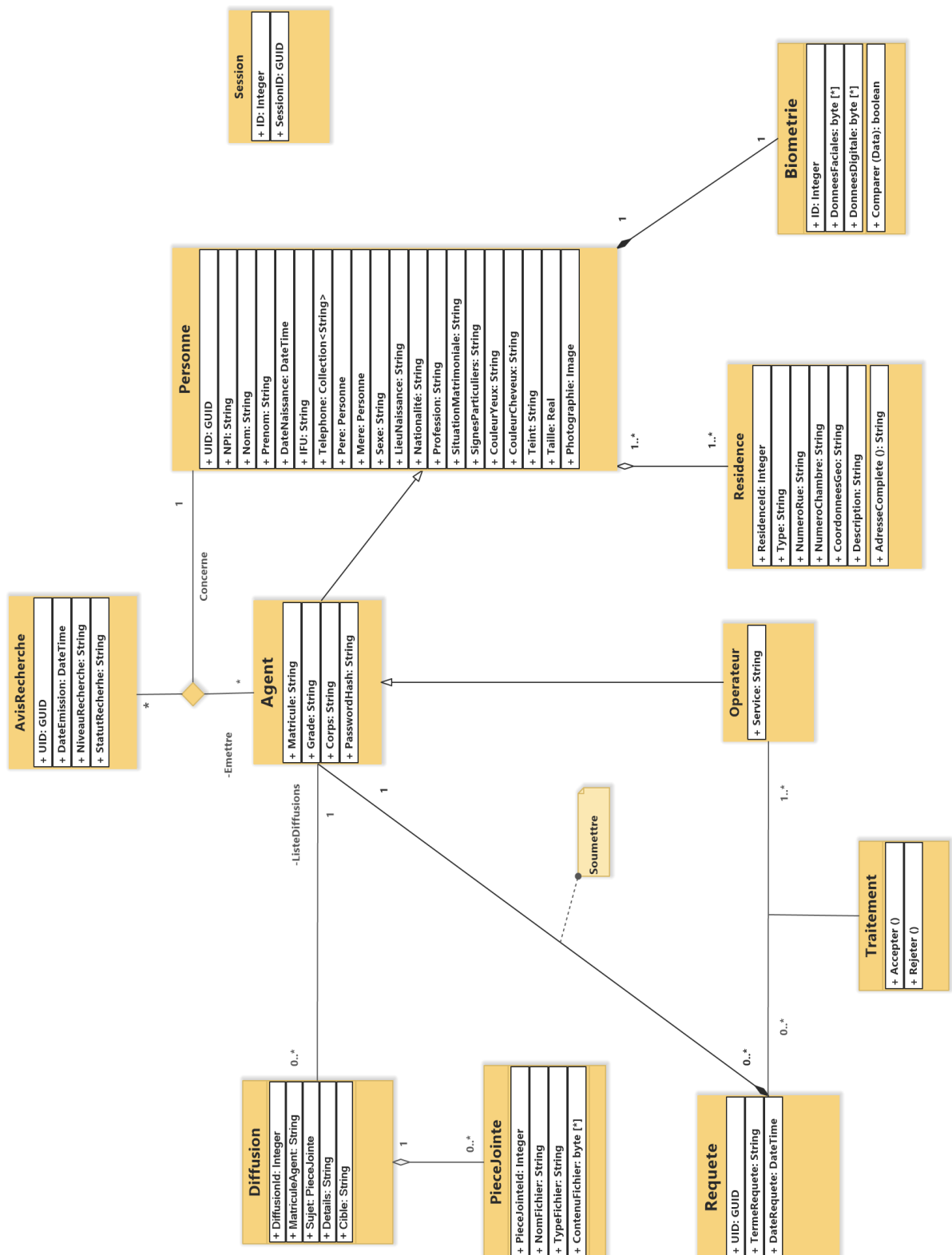
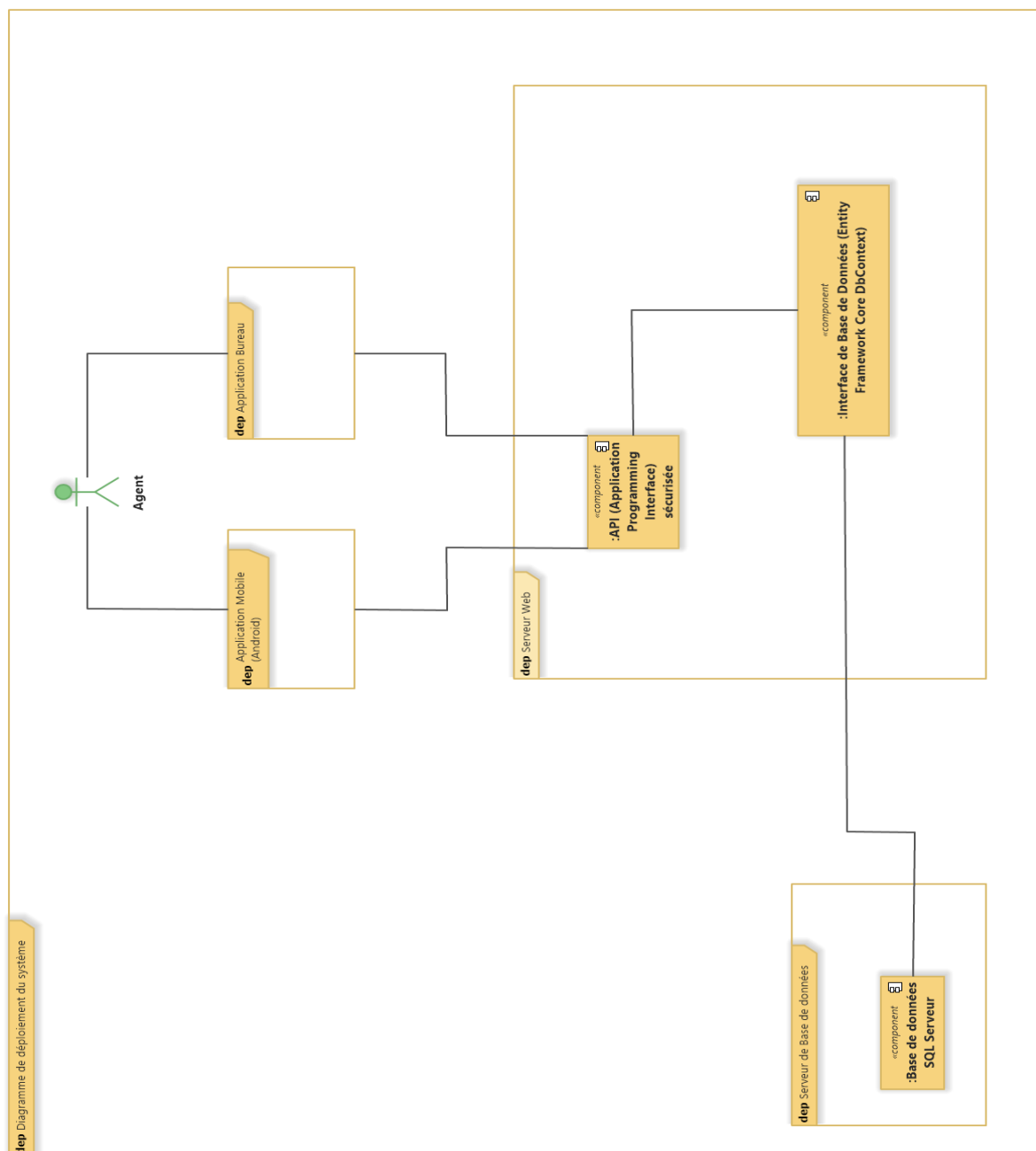


Diagramme 6: Diagramme de classes du système

2.4 Diagramme de déploiement

Le diagramme de déploiement modèle et affiche la configuration des éléments matériels (appelés « Nœuds ») et affiche comment les éléments logiciels sont déployés sur ces nœuds. Il permet également de spécifier les liens logiques et physiques via le réseau utilisé. Un nœud est soit un élément matériel ou logiciel. Au vu du système API-Client que nous aurons à déployer, le diagramme de déploiement ci-dessous présente l'architecture idéale de mise en place de ses composants logiciels et matériels.



3. Modélisation Dynamique

3.1. Diagrammes de Séquence

Le diagramme de séquence permet de représenter les communications avec et au sein du système. Il reporte des interactions entre les objets sur une base temporelle des messages échangés entre les objets (internes) et avec les acteurs (externes). Voici quelques diagrammes de séquences qu'il est utile de présenter :

- Processus d'ouverture d'une nouvelle session

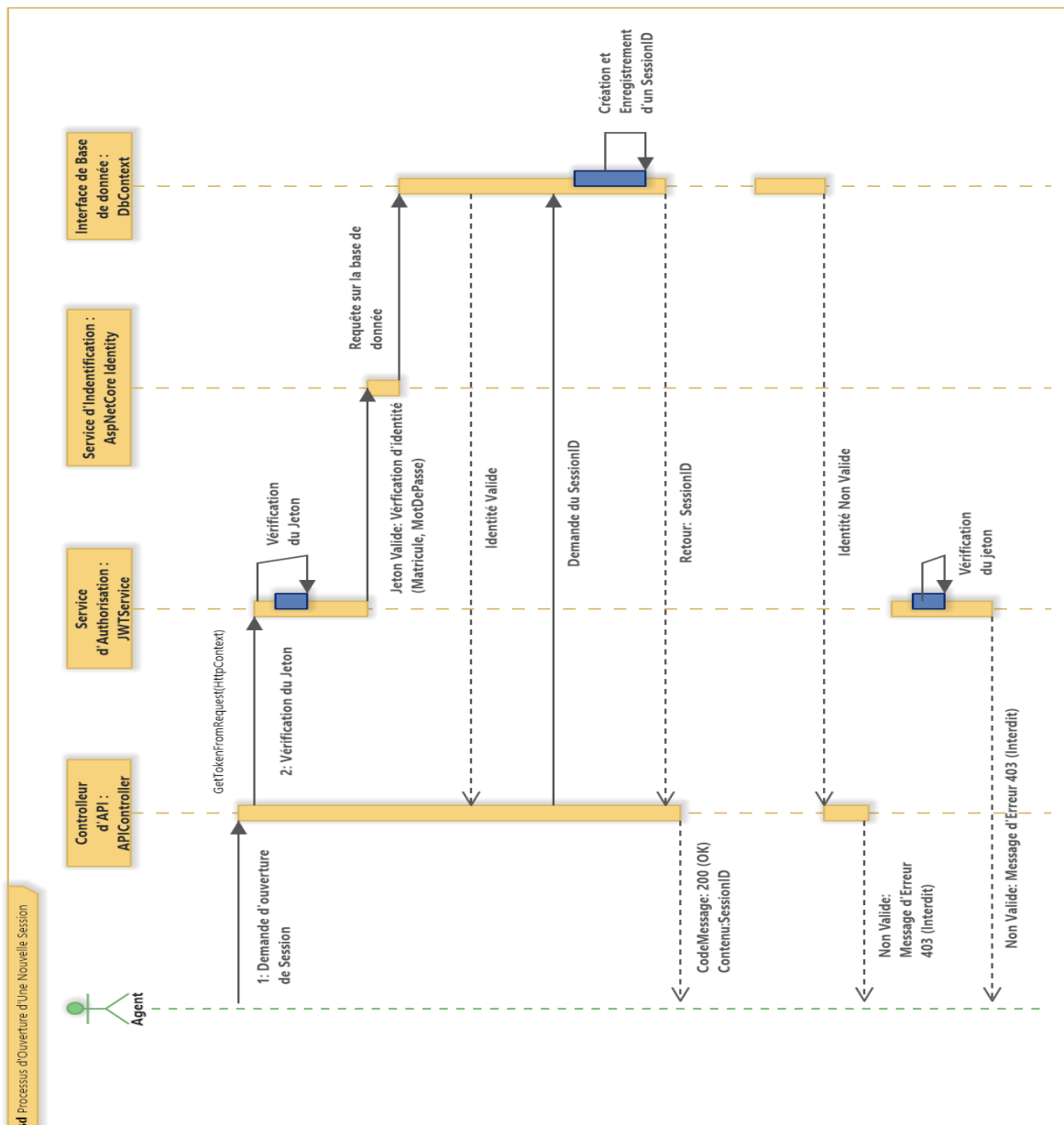


Diagramme 8: Diagramme de Séquence pour Ouverture d'une nouvelle session de travail

- Demande de la liste de tous les Agents

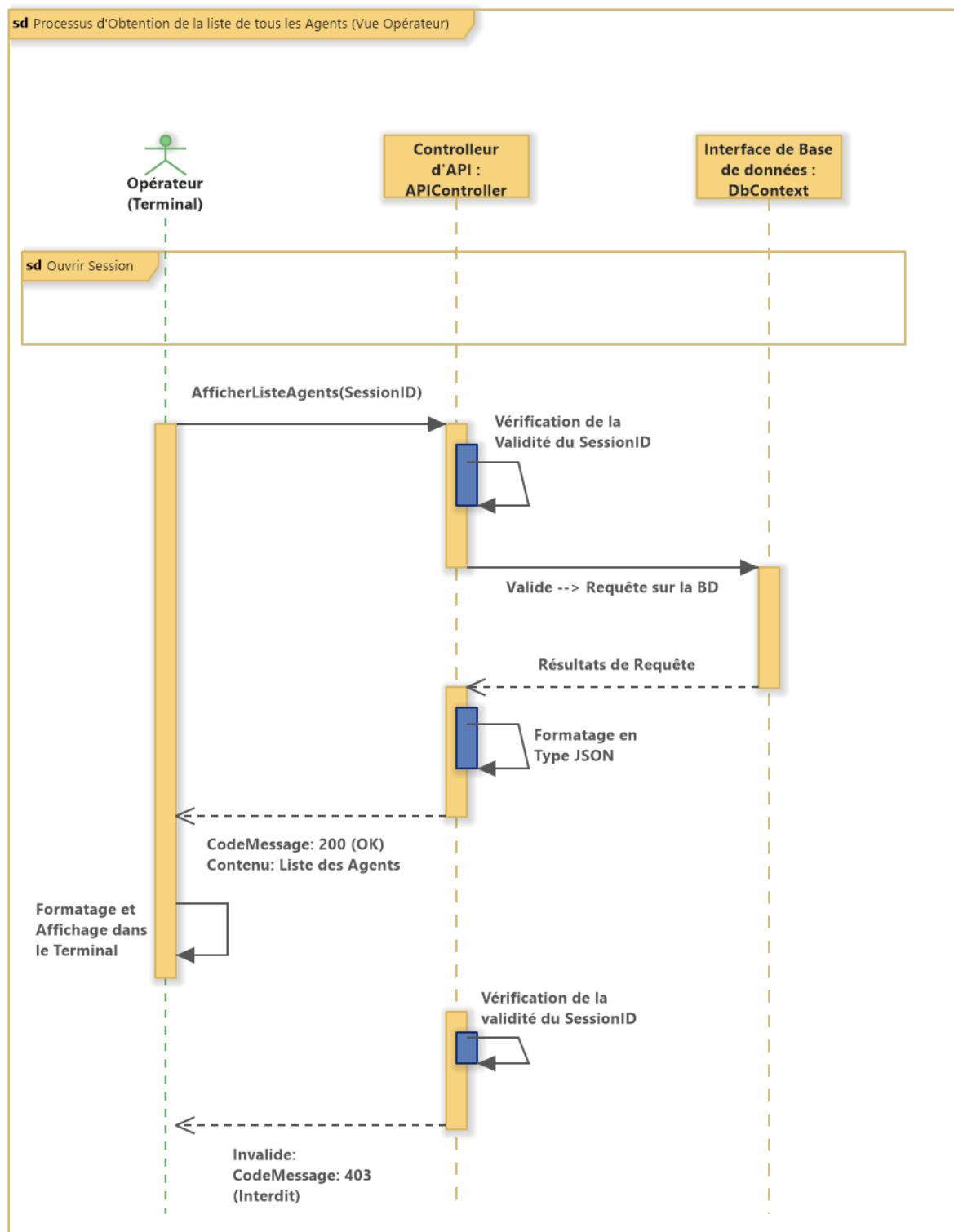


Diagramme 9: Diagramme de Séquence pour l'opération de demande de la liste des Agents

- Mise à jour d'un avis de Recherche

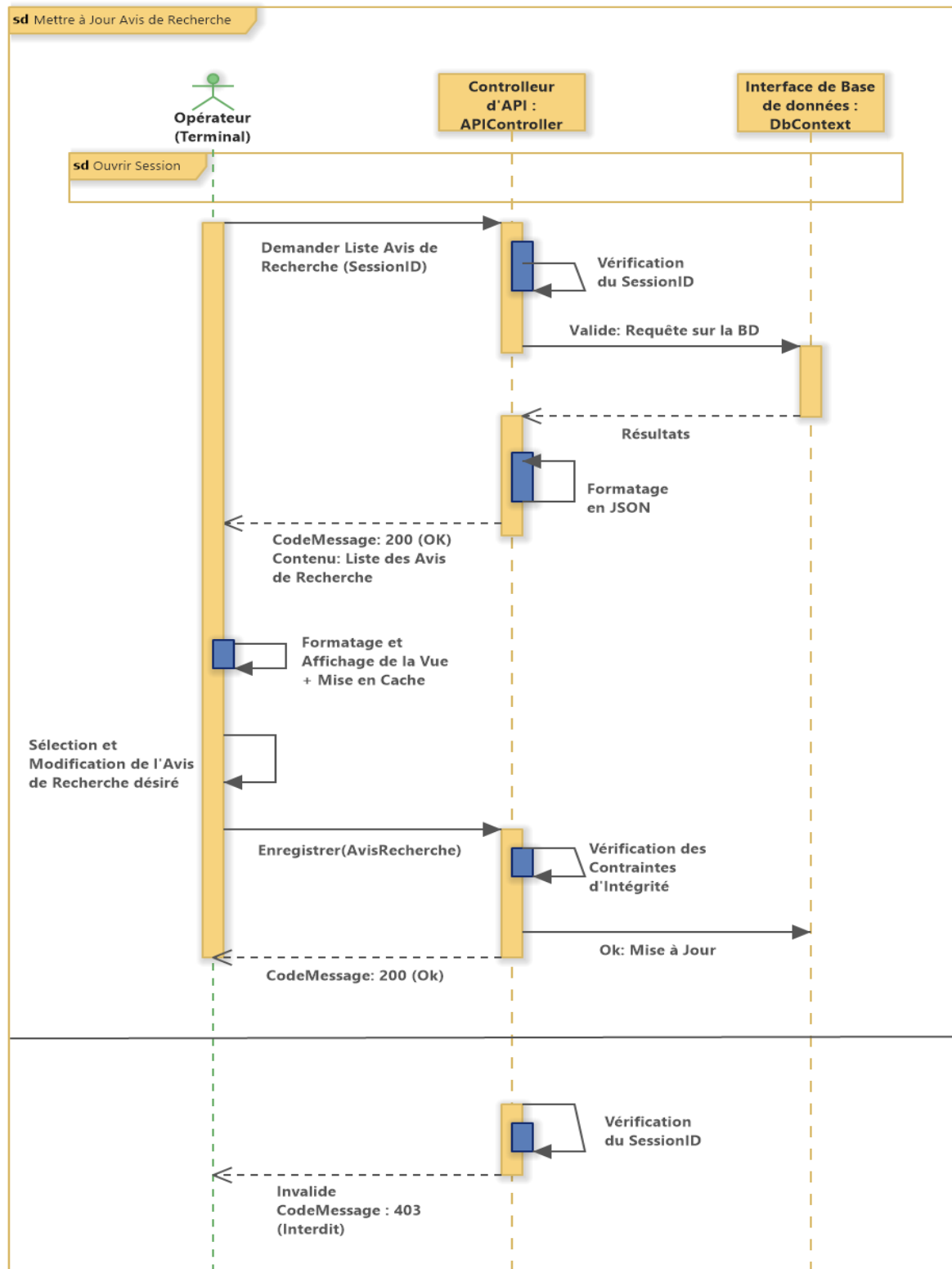


Diagramme 10: Diagramme de Séquence pour l'Opération de mise à jour d'un avis de recherche

- Chercher/Identifier une Personne

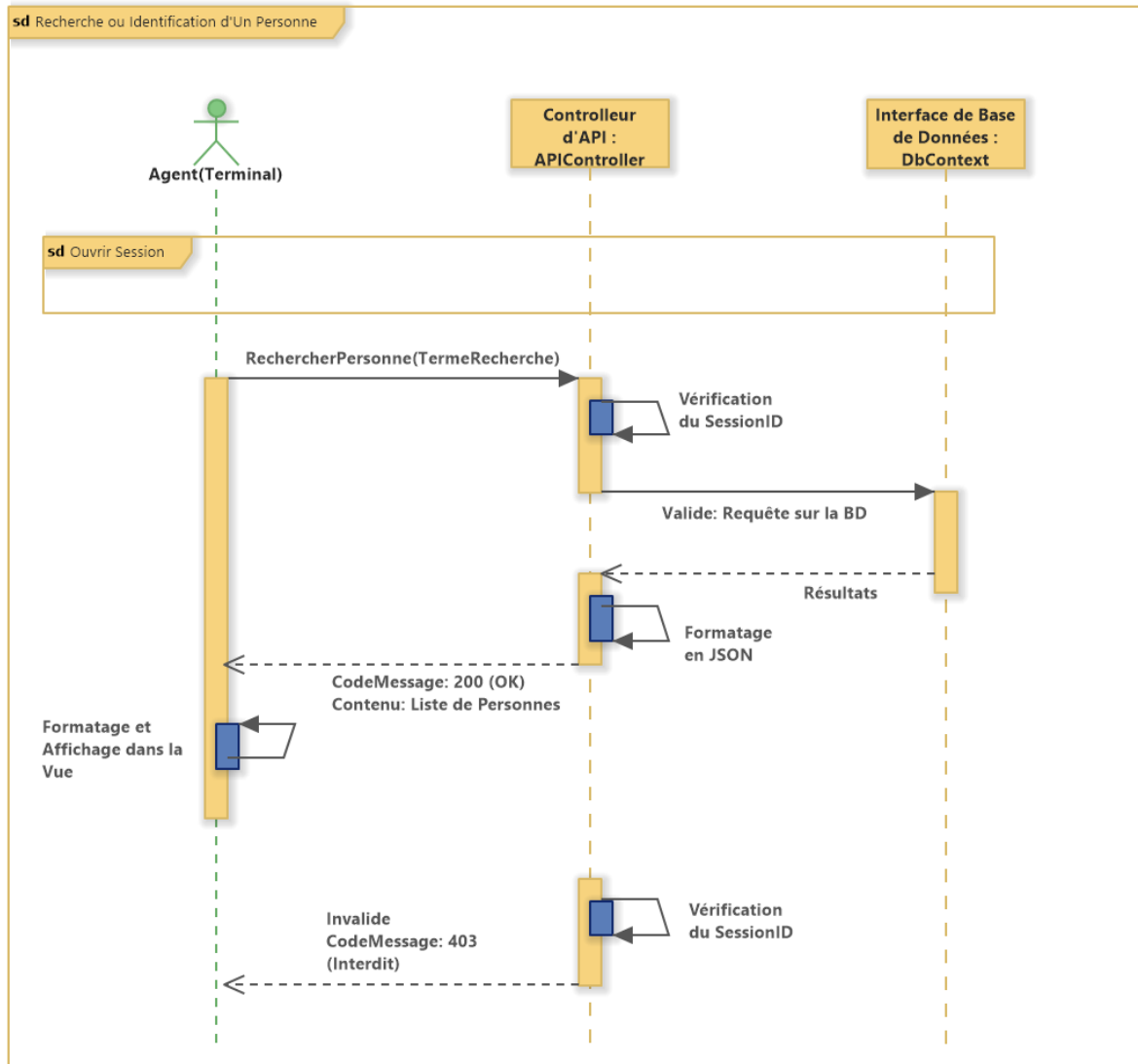


Diagramme 11: Diagramme de Séquence pour l'opération de Recherche d'identité d'une personne

- Consulter un Avis de Recherche

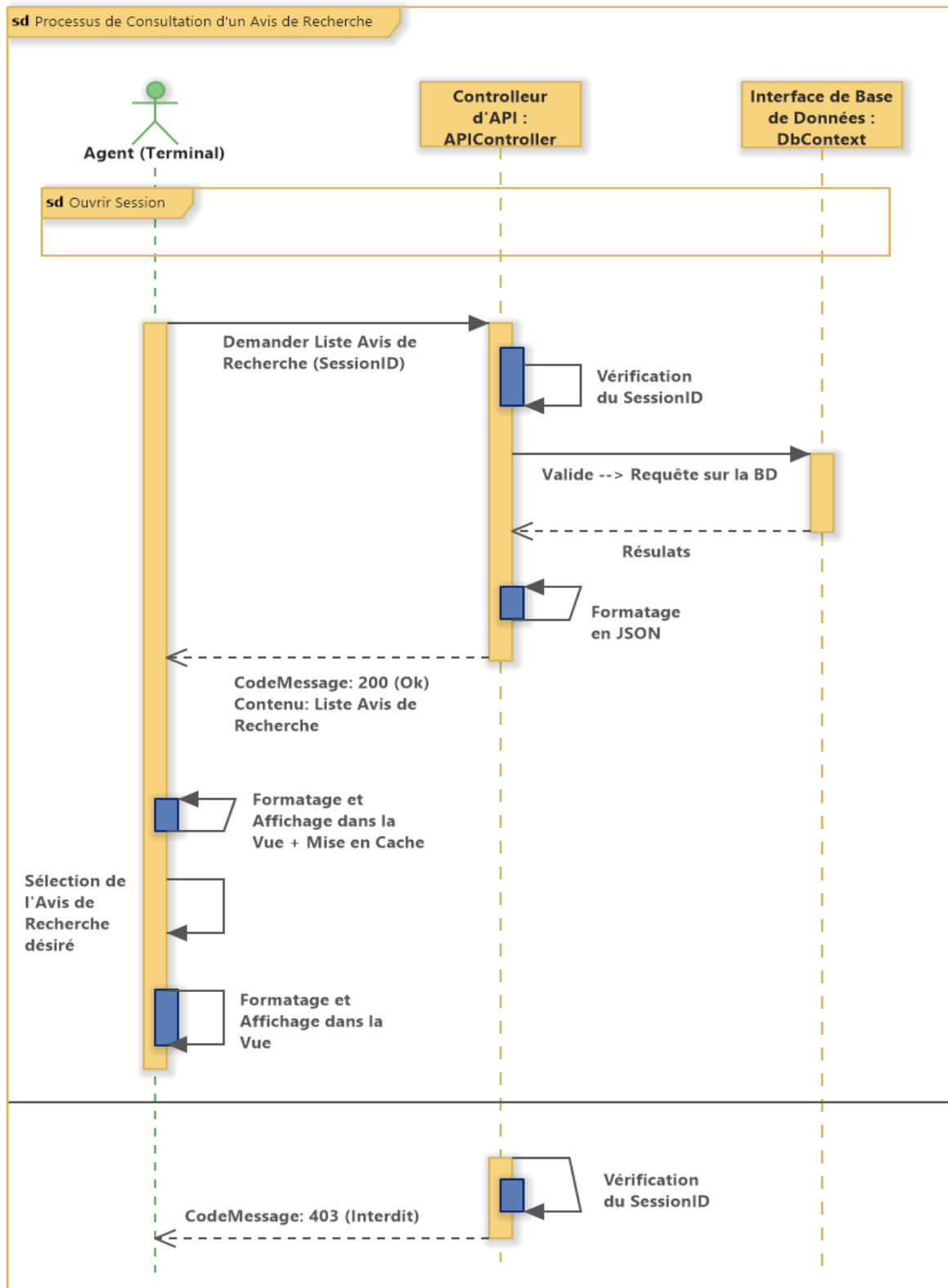


Diagramme 12: Diagramme de séquence pour l'Opération de consultation d'un avis de Recherche

CHAPITRE 3 : REALISATION DE L'APPLICATION

I. Choix et présentation des outils de développement

Dans cette section, nous présenterons les divers outils de développements que nous avons employés dans la mise en œuvre de notre projet.

Vu la taille du projet à mettre en place, nous avons pris en compte quelques considérations techniques afin de mieux pallier au problème de divergence de technologies entre les divers systèmes. Cela nous a amené notamment à choisir notre environnement ainsi que nos outils de développement. *Notre Système Comporte trois (03) projets* à savoir :

- *Une API web qui s'interface avec notre base de données, notre système d'autorisation et d'authentification pour nous permettre d'avoir accès de façon sécurisée aux données demandées ;*
- *Un Logiciel desktop qui sert de centre d'opération ;*
- *Une application Android servant de terminal (End-Point) mobile pour plus de flexibilité.*

1.1. L'environnement de Développement

Pour la réalisation et la conception d'une solution Informatique, il est très important de disposer d'un environnement de développement adapté aux besoins.

1.1.1. Le langage de développement

Avec la contrainte Orienté-Objet qu'impose la modélisation avec UML, il convient, pour le développement, d'utiliser des langages modernes, Orientés Objets et disposant d'une grande base d'utilisateurs.

Pour notre cas, un seul langage nous vient à l'esprit quand il s'agit de projets d'envergure. Il s'agit en effet du langage C# vigoureusement promu par Microsoft au cours de ces dernières décennies.



C# (C Sharp)

C# (*C Sharp*) est un langage de programmation orientée objet, commercialisé par Microsoft depuis 2002 et destiné aux développeurs de la

plateforme Microsoft.NET (une pile technologique créée par Microsoft dont le cœur est le Framework .NET).

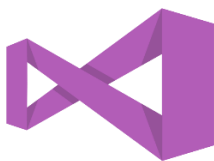
Le C# est dérivé du C++ et est très proche du Java dont il reprend la syntaxe générale ainsi que les concepts tout en y ajoutant des notions telles que la surcharge des opérateurs, les indexeurs et les délégués (delegates). Il est utilisé notamment pour développeurs d'applications web sur la plateforme ASP.NET. En C# tous les types sont des objets. Le langage comporte un ramasse-miettes (Garbage Collector) et un système de gestion d'exceptions. Nous avons opté pour ce langage tout d'abord parce que nous y sommes beaucoup plus familier, ensuite parce qu'il permet de réaliser toute sorte de projets (Web, Mobile, Desktop, Cloud, IoT.. etc.) et bénéficie d'une notoriété sans cesse croissante.

Version Utilisée : 8.0 avec la plateforme .Net Core

Bien que nous ayons eu recours à d'autres langages (comme XAML, JavaScript), il ne convient pas de les mentionner puisqu'ils n'ont été utilisés qu'à titre de langages de script. Ils n'interviennent donc que ponctuellement dans le processus de développement

1.1.2. Les Logiciels

Pour garder en un seul endroit tout le code de notre solution, nous avons opté pour l'utilisation d'un IDE (Integrated Development Environment) complet qui, non seulement nous permettra de grandement réduire notre temps de développement, mais aussi de maintenir, déboguer, construire et déployer notre projet. Il existe plusieurs IDE sur le marché, mais le plus adapté au langage que nous avons choisi est sans doute Microsoft Visual Studio.



Microsoft Visual Studio Community 2019

Visual Studio est un ensemble complet d'outils de développement permettant de développer des applications bureautiques, des applications web ASP.NET, des services web XML, et des applications mobiles. Les langages supportés (Visual Basic, Visual C++, Visual C#) utilisent tous le même environnement de développement intégré (EDI), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du

Framework .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications en tous genres.

Visual Studio Community (2019) est une version gratuite complète destinée notamment au profil «Etudiant» et équivaut, en matière de fonctionnalités, à Visual Studio Professionnel (version payante). Il faut noter que Visual Studio est unanimement désigné comme étant le meilleur Environnement de Développement Intégré au Monde.

Langages Supportés : C#, C, C++, Visual Basic, F#, JavaScript, TypeScript, Extensible Markup Language, CSS, HTML et Python.

Version Utilisée : VS 2019 (16.9.4) Community

1.1.3. Le SGBD

La persistance des données est l'une des plus délicates questions qui survient au cours du développement d'un logiciel. Cela soulève la problématique du choix d'un Système de Gestion de Base de Données "SGBD" adapté non seulement aux besoins actuels, mais qui est entièrement « scalable » afin de permettre de faire évoluer la Base de Données sans pour autant avoir à tout recommencer (ce qui pourrait induire la perte de données) .

Pour notre Projet, nous avons opté pour SQL Server Express 2014 (Pour les tests) et MySQL pour la production.



MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire et Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server. Depuis mai 2009, son créateur Michael Widenius a créé MariaDB (Maria est le prénom de sa deuxième fille) pour continuer son développement en tant que projet Open Source.

Version Utilisée : 8.0.23 avec le moteur MariaDB

1.2. L'environnement d'Exécution

L'un des points les plus importants qui vient soutenir l'environnement de développement est sans doute celui d'Exécution

1.2.1. Le Système d'Exploitation

Nous avons choisi Microsoft Windows 10 x64 dans sa Build 1909 pour le développement de notre système.

1.2.2. Le Cadriciel

Le Cadriciel de base de Windows plus connu sous le nom du dotNet Framework est un excellent point de départ pour concevoir notre système. Plusieurs considérations nous ont amené à choisir le plus récent des Cadriciel Windows disponible. Il s'agit de la plateforme dotNet Core. Un Framework totalement Open Source, léger, rapide et extensivement configurable et extensible. Il est développé par Microsoft et toute la communauté Open Source.



.NET Core Framework

.NET Core (dotNet Core) est un cadriciel (Framework) Libre et Open Source pour les systèmes d'exploitation Windows, MacOS et Linux. Il comprend CoreCLR, un environnement d'exécution complet de CLR (Common Language Runtime), la machine virtuelle qui gère l'exécution des programmes dotNet. CoreCLR est livré avec un compilateur à la volée amélioré, appelé RyuJIT et de l'environnement d'exécution CoreFX. Bien que dotNet Core partage un sous-ensemble d'API avec dotNet Framework, il est livré avec sa propre API qui ne fait pas partie du dotNet Framework. L'interface de ligne de commande du dotNet Core offre un point d'entrée d'exécution pour les systèmes d'exploitation et fournit des services de développement, tels que la compilation et la gestion des packages; ce qui permet notamment de créer divers type d'applications en utilisant n'importe quel éditeur de texte. C'est le successeur du très populaire .Net Framework.

Le Framework .Net Core supporte plusieurs langages à savoir: C#, F#, Visual Basics .Net

Version utilisée : 3.1 LTS (Long Term Support)



ASP .NET Core

ASP.NET Core est un populaire Framework Web gratuit et open-source, développé par Microsoft. Il est plus performant qu'ASP.NET. C'est un Framework modulaire qui fonctionne à la fois avec le Framework .NET (sous Windows) et .NET Core en mode multiplateforme. Cependant, depuis la version 3 du dotNet Core, ASP.NET Core fonctionne uniquement avec .NET Core abandonnant ainsi le support du Framework .NET.

Le Framework est une réécriture complète qui unifie ASP.NET MVC et ASP.NET API Web en un seul modèle de programmation.

Bien qu'il s'agisse d'un nouveau Framework, reposant sur une nouvelle pile Web, il présente un degré élevé de compatibilité avec ASP.NET MVC ce qui permet une prise en main relativement aisée. Les applications ASP.NET Core prennent en charge l'exécution simultanée de différentes applications s'exécutant sur le même ordinateur et qui ciblent différentes versions d'ASP.NET Core. Cela n'est pas possible avec les versions précédentes d'ASP.NET.

ASP Net Core permet de développer quatre (4) types d'applications Web :

- Les Web Apps
 - *ASP Net Core Razor Pages* : pour les applications web rapides avec une approche de développement centrée sur les pages Web dans un modèle freestyle.
 - *ASP Net Core MVC* : pour créer des applications web puissantes avec une approche de conception basée sur l'architecture MVC (Model-View-Controller).
 - *Blazor* : pour concevoir les Applications à Page Unique (APU)
- Les Web API Apps
 - *ASP Net Core Web API* : pour concevoir des services web (API) qui sont destinées à alimenter une large gamme de clients incluant les Navigateurs Web et les Appareils Mobiles.
- *Les Real-time Web Apps* : qui utilisent la technologie *SignalR* afin de concevoir des systèmes Temps réel.

- Les Remote Procedure Call Apps.

La partie du Framework qui nous concerne est ASP Net Core Web API. Ce sous-ensemble de ASP Net Core nous permettra d'échafauder un modèle d'API que nous personnalisons par la suite notamment en ajoutant les contraintes d'Authentification et d'Autorisation.

1.3. Autres Outils

Pour échanger les données entre les Applications et l'API, il convient d'utiliser un format de sérialisation de données adéquat afin de pouvoir récupérer ou traiter ces données. Deux formats historiques sont les plus utilisés. Il s'agit du XML et du relativement plus récent JSON. Nous choisissons JSON à cause de sa verbosité moins dense que celle du XML et de la facilité de lecture de celui-ci par rapport au XML.

JSON

JavaScript Object Notation (JSON) est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée comme le permet XML par exemple. Des bibliothèques pour le format JSON existent dans la plupart des langages de programmation.

Un document JSON comprend deux types d'éléments structurels :

- Des ensembles de paires « nom » ou « clé » / « valeur » - (clé, valeur).
- Des listes ordonnées de valeurs.

Ces mêmes éléments représentent trois types de données :

- Des objets
- Des tableaux
- Des valeurs génériques de type tableau, objet, booléen, nombre, chaîne de caractères ou null.

Bibliothèque JSON utilisée : Newtonsoft.Json v12.0.3

Afin d'implémenter le processus d'Autorisation par jetons signés, nous avons opté pour la technologie JWT que nous avons découvert dans nos recherches et qui convient idéalement au schéma que nous implémentons.



JWT

JSON Web Token (JWT) est un standard ouvert qui permet l'échange sécurisé de jetons (tokens) entre plusieurs parties. Cette sécurité de l'échange se traduit par la vérification de l'intégrité des données à l'aide d'une signature numérique. Elle s'effectue par l'algorithme HMAC ou RSA. Un jeton se compose de trois parties :

- ✓ Un en-tête (header), utilisé pour décrire le jeton. Il s'agit d'un objet JSON.
- ✓ Une charge utile (Payload) qui représente les informations embarquées dans le jeton. Il s'agit également d'un objet JSON.
- ✓ Une signature numérique.

La spécification de JWT propose différents champs (ou paramètres) standards et Optionnels, appelés Claims. Ces paramètres permettent simplement de définir plus précisément un jeton et de renforcer sa sécurité (exemple: en limitant la durée de vie d'un jeton).

Nom du paramètre	Signification
iss	Créateur du jeton
sub	Sujet du jeton
aud	Audience du jeton
exp	Date d'expiration du jeton
nbf	Date après laquelle le jeton devient invalide
iat	Date à laquelle a été créé le jeton
jti	Identifiant unique du jeton

Tableau 1: Liste des paramètres du JWT et leur signification

L'implémentation de JWT peut se faire à travers une pléthore de bibliothèques disponibles dans de nombreux langages de programmation.

Bibliothèque C# implémentant JWT utilisée: JWT.Net

Pour un projet impliquant l'usage d'une base de données, certaines considérations sont à prendre en compte lors de la mise en place de la couche d'accès aux données (DAL pour Data Access Layer). Pour faciliter l'accès aux données, nous avons décidé de faire recours à un ORM. Notre choix s'est très vite porté sur Entity Framework Core.



Entity Framework Core

Entity Framework Core est un Mappeur Relationnel-Object (Object-Relationnal Mapper en anglais) qui permet aux développeurs de la plateforme .NET de travailler avec des bases de données tout en utilisant les objets dotNet permettant ainsi d'éliminer le besoin d'écrire la plus part du code relatif à l'accès aux données (requêtes SQL ou autres types de requêtes) qui sont les plus communément écrits. C'est un outil permettant de créer une couche d'accès aux données DAL (Data Access Layer) liée à une base de données relationnelle en assurant l'indépendance du schéma conceptuel (entités ou objets) du schéma logique de la base de données, c'est-à-dire des tables.

Avec Entity Framework Core, l'accès aux données est mise en place en utilisant un modèle (les Classes représentant les entités) et d'un Contexte de base de données (Database Context) qui représente une session avec la base de données.

Entity Framework Core supporte les approches de développements suivants :

- ✓ Code First (Créer une base de données à partir du code du Modèle)
- ✓ Database First (Générer un modèle à partir d'une base de données Existante)

Les requêtes sur la Base de Données sont exécutées en utilisant LINQ (Language Integrated Query).

Version Utilisée : EF Core 3.1

La place de l'interface utilisateur dans l'ergonomie d'utilisation d'un Logiciel est trop importante pour que nous la négligions. Nous avons, Opté pour le Framework Graphique sans-égal WPF (Windows Presentation Foundation) aux fins de concevoir l'Interface de l'application Desktop de notre projet.



WPF

Windows Presentation Foundation (WPF) est un Framework graphique open source et gratuit développé par Microsoft pour assurer le rendu des applications graphiques sur Windows. Il intègre le langage descriptif XAML (eXtensible Application Markup Language) qui permet de l'utiliser d'une manière proche d'une page HTML pour les développeurs. Il est une surcouche logicielle à DirectX, remplace Windows Forms et est entièrement vectoriel pour le dessin comme pour le texte. Cela permet d'augmenter la taille des objets en fonction de la résolution de l'écran sans effet de pixellisation, il optimise également fortement la virtualisation des applications par la réduction des informations à faire transiter sur le réseau.

Il y a séparation entre les données et leur présentation, les deux aspects étant traités par WPF. WPF gère les bases de données pour les applications ou le web et fournit des modèles de présentation.

Xamarin.Forms + Open JDK et Android SDK

Qu'est-ce que Android ?

Android est un système d'exploitation mobile fondé sur le noyau Linux et développé par Google. C'est une pile de logiciels destinés à fournir une solution clé en main pour les appareils mobiles (Smartphones et Tablettes tactiles). Dans le but de concevoir la partie mobile de notre application afin d'acheminer les informations vers les agents disposés sur le terrain, Nous avons fait recours à la plateforme Xamarin pour sa conception.

Xamarin est un outil de développement « cross-platform » détenu par Microsoft depuis 2016 qui permet de générer des versions natives iOS, Android et Windows Phone d'une même application tout en utilisant un seul langage de programmation (C# ou F#) ainsi qu'une seule base de code partagée. Son implémentation Xamarin.Forms permet d'utiliser des sous-ensembles de contrôles portables mappés aux contrôles natifs d'Android, iOS et Windows Phone dans le but d'unifier toutes les précédentes déclinaisons de Xamarin (Xamarin.Mac, Xamarin 2.0 etc...).

Version utilisée : 3.0

Le « versionning » est l'une des importantes étapes du cycle de vie de développement d'une application. Pour gérer les versions au sein de notre projet, nous avons opté pour git.



Git

Git est un logiciel libre et gratuit de gestion de versions décentralisé créé par Linus Torvalds, auteur du noyau Linux. Git ne repose pas sur un serveur centralisé, mais il utilise un système de connexion pair à pair qui implique que le code informatique développé est stocké non seulement sur l'ordinateur de chaque contributeur du projet, mais peut également l'être sur un serveur dédié. C'est un outil de bas niveau, qui se veut simple et performant, dont la principale tâche est de gérer l'évolution du contenu d'une arborescence. Les serveurs Git utilisent par défaut le port 9418 pour le protocole spécifique à Git. Les protocoles HTTP, HTTPS et SSH (et leurs ports standards) peuvent aussi être utilisés.

En 2016, il s'agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes

Version utilisée : 2.29.2.3 (64 bits)



Xampp

XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (Apache-MariaDB-Perl-PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus. Officiellement, XAMPP permet de configurer un serveur de test local avant la mise en œuvre d'un site Web.



IIS (Internet Information Services)

Internet Information Services, anciennement Internet Information Server, communément appelé IIS, est un serveur Web (HTTP) des différents systèmes d'exploitation Windows NT.

IIS prend en charge plusieurs techniques Web telles les CGI, les ASP (Active Server Pages), les ASP .NET et une API (Application Programming Interface) spécifique à IIS permettant de créer des extensions et des filtres. IIS prend aussi en charge le langage PHP en mode CGI ou ISAPI.

IIS n'a été utilisé que dans l'environnement de développement grâce à sa facilité de configuration pour Visual Studio et son Support Natif des Technologies web ASP, ASP .NET et ASP .NET Core.

Version utilisée : IIS Express 10.0

II. Présentation du prototype

Une fois l'étude du système accomplie et les outils de développement déterminés, nous sommes passés à la phase de développement pour concevoir notre solution.

Comme spécifié plus haut, **notre Système Comporte trois (03) projets** à savoir :

- **Une API web** qui s'interface avec notre base de données et notre système d'autorisation et d'authentification pour nous permettre d'avoir accès de façon sécurisée aux données demandés.
- **Un Logiciel desktop** qui sert de centre d'opération
- **Une application Android** servant de terminal (End-Point) mobile pour plus de flexibilité.

Voici ci-dessous quelques Captures d'écran illustrant divers interfaces applicatives du système que nous avons conçu :

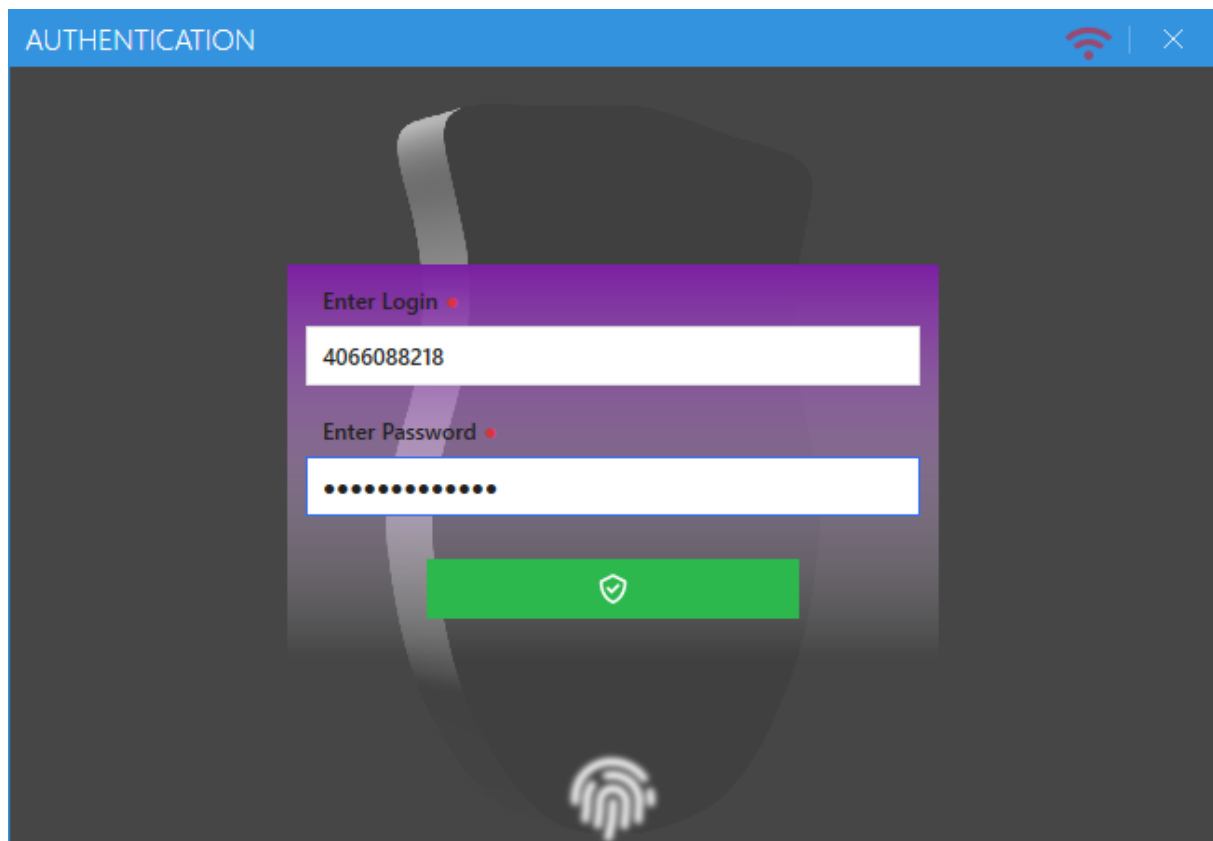


Image 1: Interface de Connexion / Authentification

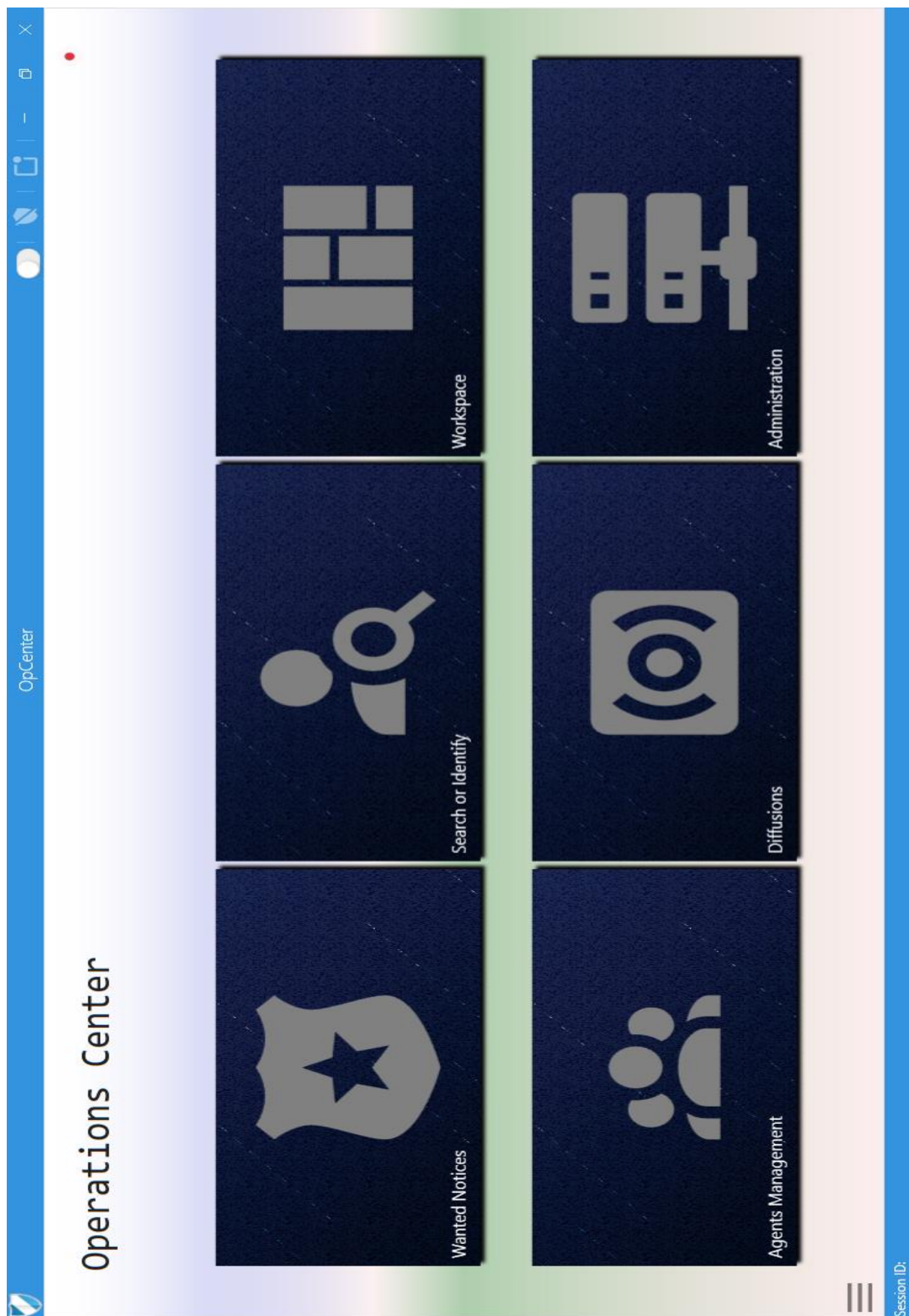


Image 2: Interface du Centre d'Opérations (Vue Développeur, Utilisateur Invalide)

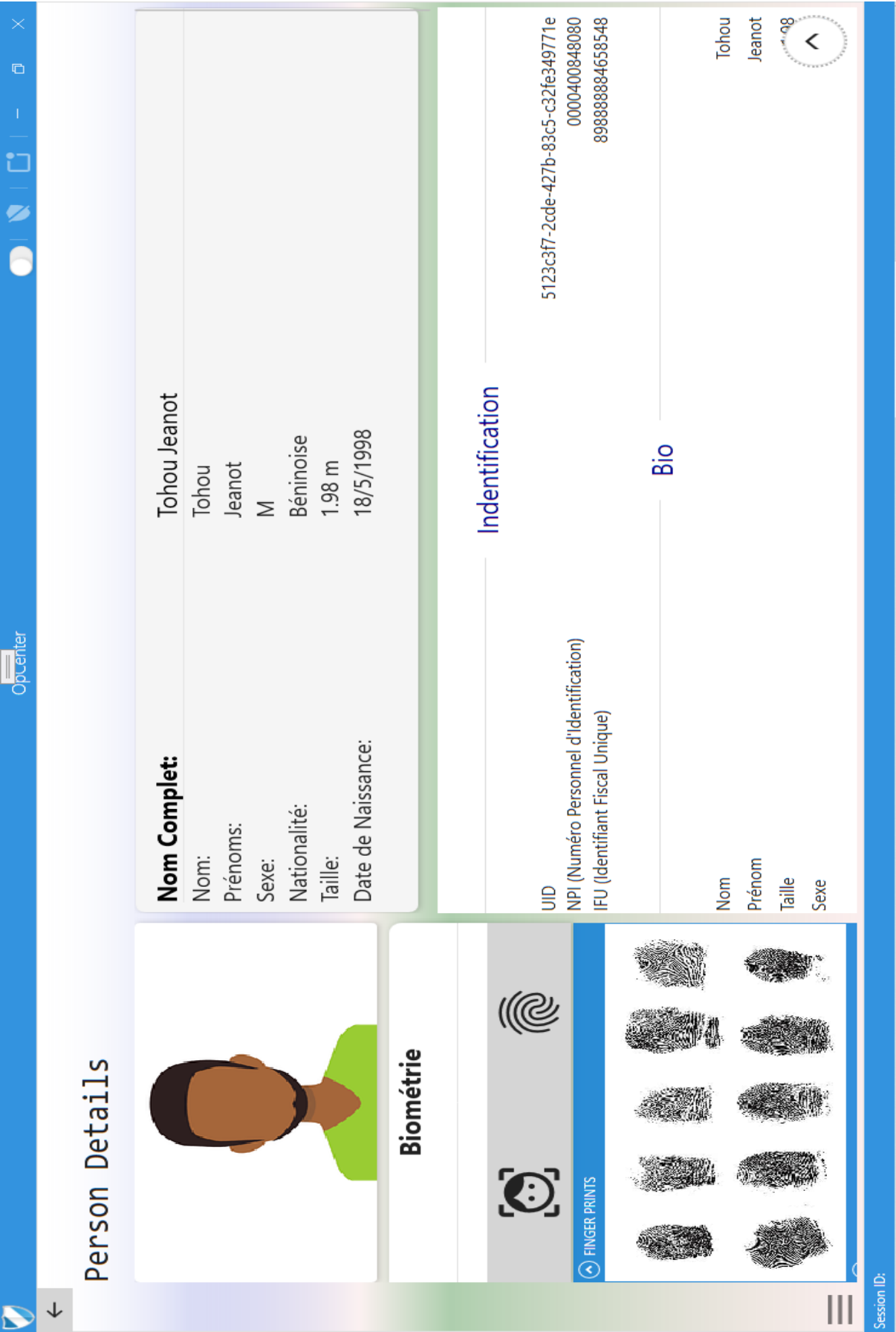


Image 3: Interface de vue sur les détails d'une personne physique

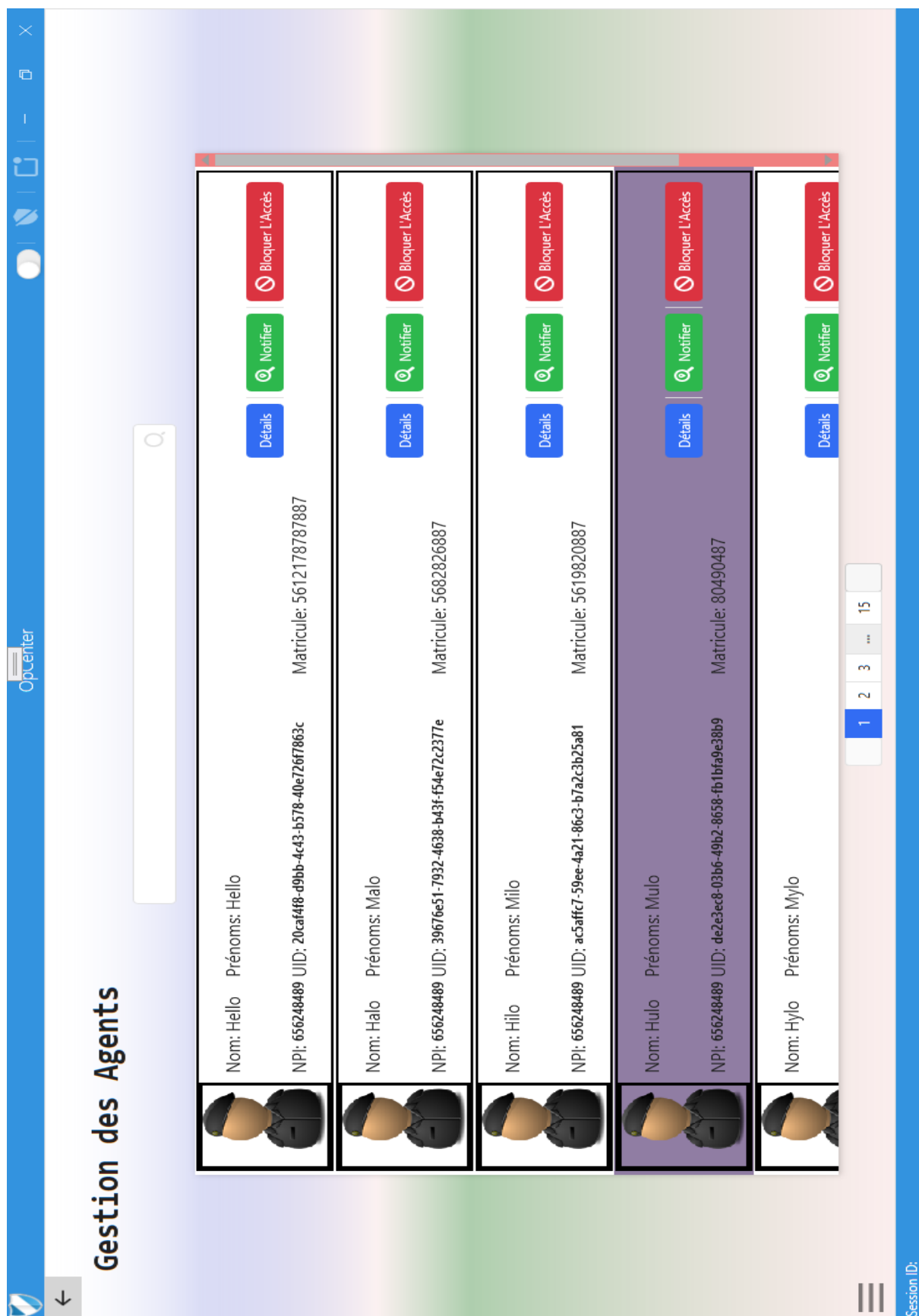


Image 4: Interface de vue sur la Gestion des Agents

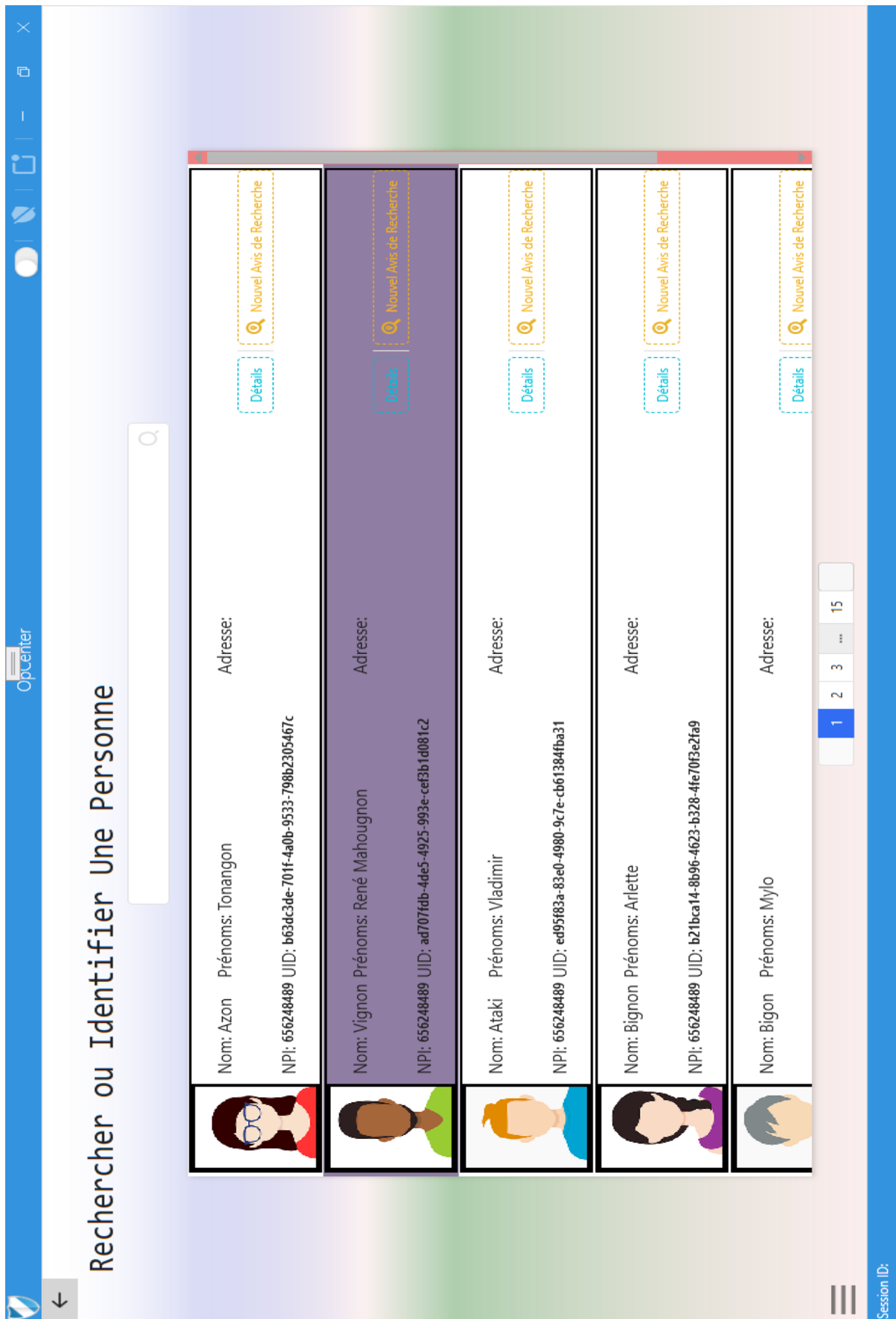


Image 5: Interface de vue sur la Recherche ou l'identification d'une Personne

III. Mesures de sécurité de l'application

Personne ne conteste aujourd'hui le fait que de nombreuses applications logicielles soient essentielles, en particulier celles qui reposent sur des technologies Internet. De simples canaux de manipulation de données statiques au départ, elles sont devenues des éléments de base complexes, dynamiques et orientés-transactions pour bon nombre d'organisations.

Le niveau de complexité et de fonctionnalité toujours plus élevé des applications logicielles actuelles a cependant engendré un effet pervers inquiétant : un nombre croissant d'organisations voient leurs logiciels être victimes d'attaques de la part d'intrus internes et externes.

La question de sécurité, bien que délicate peut être discutée selon plusieurs volets.

1. L'architecture de l'environnement d'Exécution

Le Core Common Language Runtime (CoreCLR) est le moteur qui exécute et "gère" le code d'exécution. Du point de vue de la sécurité, CoreCLR met en œuvre les restrictions d'exécution du code du .NET Core et empêche celui-ci de se comporter de manière inattendue.

Plus spécifiquement, le CoreCLR effectue une compilation juste-à-temps (JIT, Just-In-Time) lors de l'exécution du code managé. Le compilateur JIT traduit le code managé en code natif avant de l'exécuter. Étant donné que le JIT génère le code au sein du CoreCLR, ce dernier est positionné de manière unique pour en assurer la sécurité, un résultat qui ne peut pas être obtenu avec du code exécuté sans être traité dans l'environnement natif.

L'architecture de la sécurité du .NET Core Framework se compose de plusieurs éléments fondamentaux, notamment :

- **la sécurité basée sur les preuves**
- **la sécurité d'accès au code**
- **le processus de vérification**
- **la sécurité basée sur les rôles**

Dans la suite, nous ne détaillerons que les sécurités basées sur les preuves et les rôles car ce sont celles dont nous avons fait usage.

1.1 Sécurité basée sur les preuves

Les éléments clés du sous-système de la sécurité basée sur les preuves du .NET Framework incluent la stratégie, les autorisations et la preuve.

- La Stratégie

La stratégie dans .NET Core est omniprésente et bien protégée contre tout accès par les utilisateurs qui ne sont pas administrateurs. Elle est installée automatiquement sur chaque ordinateur, pour chaque compte d'utilisateur, et peut être déployée facultativement sur des domaines Windows via une stratégie de groupe.

La fonction de base de la stratégie de sécurité dans le .NET Framework consiste à associer les autorisations aux preuves (nous présenterons ces deux concepts dans un instant). Elle peut aussi être personnalisée par des comptes administratifs disposant des droits d'accès appropriés, afin de répondre à des besoins spécifiques.

- L'Autorisations

Les autorisations sont à la base de la Stratégie. Elles décrivent une ou plusieurs ressources et les droits associés, et implémentent des méthodes de demande et d'assertion d'accès. Le développeur peut étendre les définitions de ces autorisations afin d'inclure les ressources définies par l'application et les méthodes de vérification des droits d'accès. De telles personnalisations granulaires ne peuvent pas être effectuées aussi facilement dans d'autres architectures de code managé telles que Java 2. Les développeurs peuvent contrôler la manière dont leur code réagit aux autorisations accordées par la stratégie en incorporant des demandes d'autorisation dans les assemblys. Cela permet au code d'être conçu pour s'exécuter avec moins de privilèges.

On distingue trois types de demandes d'autorisation : *Minimal*, *Optional* et *Refuse*.

- Preuve

Au moment de l'exécution, le CLR détermine les autorisations qui peuvent être assignées à un assembly particulier en évaluant la preuve de cet assembly. La preuve peut provenir d'une variété de sources résidant dans un assembly ou être obtenue à partir de l'environnement d'exécution local. Parmi les sources de preuve, on peut citer :

- i. Les espaces de noms scellés de manière cryptographique (noms forts) ;
- ii. L'identité de l'éditeur du logiciel (AuthentiCode) ;
- iii. Le code d'origine (adresse URL, site, Internet Explorer Zone).

- Intégration de la stratégie, des autorisations et de la preuve

Jusqu'à présent, nous avons décrit séparément chacun des principaux composants du modèle de sécurité basée sur les preuves du .NET Framework. Toutefois, il est important de noter que ces composants interagissent parfaitement pour fournir un environnement d'exécution très différent de ceux auxquels nous sommes habitués lors de l'exécution de fichiers exécutables.

Du point de vue du développeur, la majeure partie du travail s'effectue en arrière-plan. Tant que des autorisations suffisantes¹ et des stratégies correctement configurées couvrent toutes les ressources impliquées, et tant que le développeur utilise du code managé pour y accéder, le contrôle des preuves et la mise en œuvre de la stratégie requis sont traités en toute transparence.

3.1.1. Sécurité basée sur les rôles

La sécurité basée sur les rôles définit la procédure de création d'une identité par le .NET Core Framework et la manière dont il autorise ou refuse l'accès aux ressources pour cette identité. Ces deux processus, communément appelés "authentification" et "autorisation", sont les pivots du développement d'applications sécurisées pour le Web (MVC, API, Razor Page, etc...)

- Authentification

La sécurité basée sur les rôles donne aux développeurs la possibilité de construire des scénarios d'authentification hautement personnalisés pour leurs applications. Toutes les routines d'authentification les plus courantes sont disponibles pour les applications .NET Framework par le biais d'une vaste gamme de fournisseurs d'authentification. Il s'agit en fait de routines de code qui vérifient les informations d'authentification, créent les objets Identity et Principal appropriés, et les attachent au contexte de la demande.

- Autorisation

Une fois l'identité établie de manière fiable à l'aide de l'une de ces méthodes bien connues, l'accès aux ressources peut être autorisé par le biais d'une architecture extensible et souple similaire.

Pour notre projet, Nous avons utilisé JWT comme fournisseur d'Autorisation. L'identification a été implémentée par nous même à cause du cas particulier de notre système dont les « Utilisateurs » sont également de potentiels « Administrateur ».



➤ **Vulnérabilités**

Les JWT sont un moyen d'authentification sécurisé. Il existe néanmoins certaines vulnérabilités, qui sont évitables en les considérant avec précaution. En voici une liste non exhaustive :

Un JWT peut être soit encodé soit chiffré. Un JWT uniquement encodé, comme c'est le cas de la majorité des JWT en circulation, devra donc transiter par un protocole de communication sécurisé (HTTPS) pour qu'il soit inutilisable par un éventuel espion ou utilisateur mal intentionné qui serait parvenu à l'intercepter.

La clé secrète doit être conservée de façon extrêmement sécurisée. Tout utilisateur (ou système) en possession de la clé secrète d'un serveur serait en mesure de générer des JWT reconnus comme valides par ce serveur.

➤ **Solutions que nous avons implémentées pour pallier à ce problème**

Nous avons choisi d'augmenter la sécurité en ajustant les paramètres du JWT et en définissant des paramètres propres à notre domaine afin d'effectuer toutes les vérifications supplémentaires requises.

Nom du paramètre	Valeur Client Mobile	Valeur Client Lourd (Desktop)
iss	FastLaneApp	OpCenterApp
sub	✓ Authentication ✓ Request ✓ SessionID	✓ Authentication ✓ Request ✓ Notification
aud	TerminalMobile	TerminalDesktop
exp	✓ 5 Minutes après l'émission (Authentication) ✓ Usage unique (Request)	✓ Usage unique (Request) ✓ 5 Minutes après l'émission (Authentication)
nbf		Date & Heure d'émission (en Minutes) + 1440 Minutes
iat	Now (Maintenant)	Now (Maintenant)
jti	Ceci est un Hash généré sur une clé secrète enregistrée dans la configuration des applications. La Somme de contrôle du Hash est vérifiée de bout en bout. Toute requête ne satisfaisant pas cette condition est rejetée	

Tableau 2: Tableau de convention des valeurs des paramètres jwt utilisés

En plus de ces paramètres, nous avons ajouté des paramètres (ou claims) additionnels qui pour la plus part sont soit automatiquement générés par les différentes applications, soit stockés de façon arbitraire.

- **bks** : raison supplémentaire de la requête.
- **sid** : Hash du Session ID pour toute requête d'obtention ou de post de données.
- **rol** : le rôle de l'utilisateur présent.

Exemple de jeton calculé sur la base de paramètres de minimaux :

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJleHAiOiJlY2MjI3MDAxNDQsImZcy6llbVbGljZU9wQVBjIiwiaXVkljoieGVGyYbWluYWxzZXNrdG9wliwibWF0IjoieODk4OTg1OTgiLCJwd2QiOiI3N2E4emVlYTg3IiwiaYmtzIjoieU2Vzc2lubiJ9.e4aNT2O0XI05jWz2T2z4Nvedvm0eMBY0gIM-ar3C42A

2. La Cryptographie

Tout au long du processus d'encodage, de décodage, de cryptage et de décryptage, plusieurs concepts cryptographiques ont été mise en œuvre pour contrer le piratage des données sensibles.

- L'encodage: Tout simplement, c'est le processus de production d'un message selon un code.
- La cryptographie quant à elle est une technique d'écriture où un message chiffré est écrit à l'aide de codes secrets ou de clés de chiffrement. La cryptographie est principalement utilisée pour protéger un message confidentiel.

Pour nos **encodages** simples, nous avons à plusieurs reprises employé l'encodage en **Base64url**.

Pour diverses **opérations cryptographiques**, nous avons employé deux (02) des nombreux algorithmes de Cryptographie/ Hachage proposés par la spécification JWT. Il s'agit du **HMAC + SHA256** (Symétrique) et du **RSASSA-PKCS1-v1_5 + SHA256** (Asymétrique).

- Le codage en Base64url

C'est une variante du codage en Base64 qui permet d'encoder l'information utilisant 64 caractères choisis pour être disponibles sur la majorité des systèmes. Il est principalement utilisé pour la transmission des messages sur internet.

- **Inconvénients** : ce codage augmente la taille des données dans le cas où elles sont des données de type textuelles
- **Avantages** : L'intérêt du base64url ne se retrouve donc pas que dans la représentation des données textuelles, mais surtout dans la représentation des données binaires. *Pour preuve, nous n'avons utilisé le base64url que pour stocker nos données binaires (Images – photographies - données d'empreinte digitale et données faciales en base de données). Le décodage permet de récupérer ces données.*

- Les Algorithmes de cryptographie/hachage

Les deux principaux que nous avons employés pour notre projet sont le *HMAC + SHA256 (Symétrique) pour les tests et le RSASSA-PKCS1-v1_5 + SHA256 (Asymétrique) pour la version release (actuellement en bêta – grâce au versionning).*

HMAC + SHA256 : Il s'agit d'une spécification permettant de signer les messages en employant une fonction de hash cryptographique. Plus difficile est ce hash, plus fort la sécurité du message signé généré. La fonction de hash utilisé ici est le SHA256.

RSASSA-PKCS1-v1_5 + SHA256 : Ici, l'algorithme de signature numérique utilisé est le RSA. La fonction de hash utilise le SHA256. Notons que cet algorithme nécessite la mise en place d'un certificat puis qu'il est Asymétrique.

3. Secure Socket Layer (SSL) / TLS (Transport Layer Security)

SSL était le protocole cryptographique le plus largement utilisé pour assurer la sécurité des communications sur internet avant d'être aujourd'hui remplacé par le protocole TLS. Le terme « SSL » est, de nos jours, utilisé pour faire référence à ce type de technologie. SSL Crée un canal sécurisé entre deux machines communiquant sur un réseau interne ou Internet. L'adresse des machines passe alors du http au HTTPS, le « S » signifiant Sécurisé.

Pour notre projet, nous avons utilisé le certificat auto-signé généré par ASP .Net Core Web API pour assurer l'acheminement de l'information par un canal sécurisé SSL.

En plus de cela, ***notre API est configurée de façon à forcer la redirection de rediriger de tout le trafic HTTP sur un canal HTTPS*** de manière à s'assurer que la machine qui communique avec elle utilise une connexion sécurisée. ***Lorsque la machine ne peut fournir une connexion sécurisée, elle n'est pas autorisée à se connecter.*** Elle est implémentée dans la classe de démarrage de notre application (Startup.cs) par la ligne suivante :

```
app.UseHttpsRedirections();
```

4. La Restriction d'adresses IP (Internet Protocol)

Une mesure de sécurité très importante que nous avons décrétée est la restriction d'adresse IP. Il est inconcevable que les adresses IP des machines qui soumettent les requêtes à notre API puissent être en dehors du territoire National Béninois. Ce faisant, nous avons restreint l'accès aux adresses IP qui sont hors du territoire National Béninois.

Les restrictions auxquelles nous avons pensé ne s'arrêtent pas là, il faudrait encore limiter les requêtes des terminaux dont paramètre *issuer* (**iss**) du JWT est de type « OpCenterApp » aux requêtes adresses IP qui proviennent des postes de polices qui sont autorisés à travailler avec notre système.

Ces restrictions ont été implémentées en maintenant diverses Listes d'Adresses Autorisée que nous avons appelées:

- IP-SafeList pour le cas général
- IP-AdminSafeList pour les Administrateurs
- IP-OperatorSafeList pour les Opérateurs

Nous avons donc utilisé un Middleware fonctionnant en amont du « Process » principal (tout simplement notre classe de démarrage) qui charge, vérifie et autorise les requêtes conformément aux restrictions indiquées.

5. Authentification par empreinte digitale

Cette fonctionnalité n'est prévue pour l'instant qu'avec l'application Mobile afin d'authentifier l'Agent-utilisateur. L'authentification normalement prévue par défaut précède naturellement l'authentification par empreinte digitale. Il s'agit d'une Authentification à deux Facteurs (2FA : Two Factors Authentication)

CONCLUSION

En somme, dans ce document nous avons rapporté le travail effectué au cours de notre stage au sein de l'entreprise KONECT Technologies. Ce travail avait pour but la conception et la réalisation d'un système de contrôle d'Identification pour la police.

Nous avons donc pour objectif la conception d'un système fonctionnel sécurisé de fourniture et d'accès à l'information aussi bien sur PC que sur téléphone Intelligent (Smartphone Android) qui puisse apporter une solution quant aux problèmes rencontrés par la police nationale à savoir l'impossibilité d'identification avec uniquement les informations fournies par un potentiel suspect, l'impossibilité d'identification en tout lieu et le manque d'un système mobile adéquat pour l'échange d'informations. Pour parvenir à nos fins, nous avons utilisé UML comme méthode d'analyse de notre projet, le SDK Net Core qui permet de développer des applications multiplateformes, Visual Studio Community 2019 comme IDE, C# comme langage de programmation, ASP Net Core pour Framework de développement d'API et MySQL pour la base de données et Xamarin pour l'application sous Android.

A la fin de notre stage, nous pouvons noter des points de satisfaction quant au travail réalisé ainsi que l'ensemble des connaissances que nous avons pu assimiler tout au long du stage. Notons que nous avons pu réaliser l'application pour PC ainsi que celle pour Smartphone Android et effectuer des tests qui se sont montrés très concluants. Le stage nous a permis de renforcer nos connaissances quant à la consommation des API, la mise en place d'un système d'authentification, d'encodage-cryptage sécurisé, de nous familiariser avec le développement sous Android ainsi que le développement multiplateforme.

Vu l'importance du projet nous jugeons nécessaire qu'on prenne en compte d'autres moyens de déploiement comme la boucle locale radio afin d'utiliser à la place d'Internet, les ondes radioélectriques comme support de transmission de l'information. Aussi en lieu et place de terminaux fonctionnant sous Android, l'on pourrait envisager des appareils plus sécurisés dédiés au système

REFERENCES BIBLIOGRAPHIQUES

- JWT HandBook v0.14.1 disponible au format PDF
Consulté le 31/3/2021
- Loi N° 2017-08 du 19 Juin 2017 portant identification des personnes physiques en République du Bénin.
Consulté le 04/05/2021
- DECRET N° 2020 - 187 DU 11 MARS 2020 portant autorisation de collecte et de traitement par la Police républicaine des données personnelles des voyageurs au niveau des frontières du Bénin.
Consulté le 04/05/2021
- La sécurité dans Microsoft .NET Framework : Analyse par Foundstone, Inc. et CORE Security Technologies.
Consulté le 26/05/2021
- SIPAO Programme
Consulté le 16/04/2021
- Stratégie de formation du programme SIPAO
Consulté le 16/04/2021
- Les bonnes pratiques en matière de protection des données à caractère personnel
Consulté le 16/04/2021
- Lettres d'information SIPAO N°1 à N°16
Consulté le 16/04/2021
- ProjectSheet – Programme SIPAO
Consulté le 16/04/2021

WEBOGRAPHIE

- Programme SIPAO

<https://interpol.int/fr/Notre-action/Renforcement-des-capacites/PROGRAMME-SIPAO>

- Web API

<https://faun.pub/restful-web-api-using-c-net-core-3-1-with-sqlite-f020d76c9b89>

<https://clevindsouza.medium.com/what-is-an-api-create-a-simple-rest-api-using-php-and-mysql-1eb54a8d0afb>

<https://zestedesavoir.com/tutoriels/597/creez-votre-site-web-avec-asp-net/allons-plus-loin-avec-asp-net/les-api-rest/>

<http://sebdelaporte.fr/micro-controleur/raspberrypi/creation-dune-web-api-rest-avec-asp-net-core-mvc>

<https://rdonfack.developpez.com/tutoriels/documenter-web-api-aspnet-core-swagger/>

<https://www.tutorialsteacher.com/webapi/what-is-web-api>

- Bases de données & SQL

<https://www.slideshare.net/mobile/billkarwin/sql-injection-myths-and-fallacies>

<https://medium.com/@emekadc/how-to-implement-one-to-one-one-to-many-and-many-to-many-relationships-when-designing-a-database-9da2de684710>

<https://fr.m.wikipedia.org/wiki/MySQL>

https://fr.m.wikipedia.org/wiki/Microsoft_SQL_Server

- JWT

<https://auth0.com/resources/ebooks/jwt-handbook>

<https://jwt.io/>

- Framework

<https://fr.m.wikipedia.org/wiki/Framework>

- NoSQL

<https://robvolk.com/nosql-design-patterns-for-relational-data-9c2c11ae3b4a>

- Xamarin

<https://github.com/XLabs/Xamarin-Forms-Labs>

- Cryptage

<https://github.com/sjh37/Effortless-.Net-Encryption>

- Git

<https://www.freecodecamp.org/news/practical-git-and-git-workflows/amp/>

<https://fr.m.wikipedia.org/wiki/Git>

- ASP .NET Core

https://fr.m.wikipedia.org/wiki/ASP.NET_Core

<https://github.com/dotnet/AspNetCore.Docs/blob/main/aspnetcore/security/authentication/customize-identity-model.md>

<https://www.tektutorialshub.com/asp-net-core/asp-net-core-identity-tutorial/>

<https://github.com/domaindrivendev/Swashbuckle.AspNetCore>

<https://cdiese.fr/aspnet-core-dependency-injection/>

- OAuth

<https://zestedesavoir.com/articles/1616/comprendre-oauth-2-0-par-l'exemple/>

<https://medium.com/@dmitrysikorsky/asp-net-core-custom-user-manager-a7206e718a90>

<https://blogs.infinisquare.com/posts/productivite/tiny-restclient-le-client-rest-pour-consommer-vos-api-rest>

- WPF

<https://github.com/HandyOrg>

- Sécurité

<https://medium.com/@mmoshikoo/jwt-authentication-using-c-54e0c71f21b0>

<https://devblogs.microsoft.com/aspnet/asp-net-core-authentication-with-identityserver4/>

<https://www.ocpsoft.org/tutorials/regular-expressions/password-regular-expression/>

<https://identityserver4.readthedocs.io/en/latest/>

<https://docs.microsoft.com/en-us/aspnet/identity/overview/extensibility/overview-of-custom-storage-providers-for-aspnet-identity>

- RavenDB

<https://ravendb.net/docs/article-page/5.1/csharp>

<https://ravendb.net/articles/proper-software-architecture-for-a-new-project>

- Net Core

https://fr.m.wikipedia.org/wiki/.NET_Core

<https://medium.com/@dbottiau/advanced-realtime-streaming-with-signalr-in-net-core-2e38fce26fbb>

<https://www.c-sharpcorner.com/article/signalr-in-net-core/>

<https://fr.qaz.wiki/wiki/SignalR>

https://fr.qaz.wiki/wiki/Remote_procedure_call

- Java

[https://fr.m.wikipedia.org/wiki/Java_\(langage\)](https://fr.m.wikipedia.org/wiki/Java_(langage))

- Visual Studio

https://fr.m.wikipedia.org/wiki/Microsoft_Visual_Studio

- Entity Framework

<https://unaura.com/utiliser-mysql-avec-entity-framework-core/?lang=fr>

<https://blog.cellenza.com/dotnet/net-core/quoi-de-neuf-entity-framework-core/>

<https://docs.microsoft.com/en-us/ef/core/>

https://en.m.wikipedia.org/wiki/Entity_Framework

- XAMP & Composants

<https://www.apachefriends.org>

https://fr.m.wikipedia.org/wiki/Apache_HTTP_Server

https://fr.wikipedia.org/wiki/JSON_Web_Token

<https://fr.m.wikipedia.org/wiki/XAMPP/>

- JWT

<https://fr.m.wikipedia.org/wiki/XAMPP>

- Android

<https://fr.wikipedia.org/wiki/Android>

- REST

https://fr.m.wikipedia.org/wiki/Representational_state_transfer

- CSharp

https://fr.m.wikipedia.org/wiki/C_sharp

- JSON

https://fr.m.wikipedia.org/wiki/JavaScript_Object_Notation

- WPF

https://fr.m.wikipedia.org/wiki/Windows_Presentation_Foundation

- IIS

https://fr.m.wikipedia.org/wiki/Internet_Information_Services