

**Développement d'une application Flutter et d'un
module Odoo pour la gestion des sanctions
administratives communales**



Travail de fin d'études présenté par

Yassine EL BAKKOURY

En vue de l'obtention du diplôme de

Master en Sciences de l'Ingénieur Industriel orientation Informatique

Remerciements

Je tiens à remercier les personnes qui ont contribué au développement du projet et à la rédaction de ce travail de fin d'étude, et plus particulièrement :

Marc Pirene, co-fondateur de Noviat SA et promoteur de ce TFE, pour ses conseils, son investissement, sa disponibilité et ses idées qui m'ont aidé à avancer tout au long de l'année.

Mr. Rémy Taymans, tuteur de ce TFE, pour la pertinence de ses remarques et son suivi régulier.

Abstract

Une bonne administration publique permet de répondre correctement aux besoins de la société en permettant une prospérité économique ainsi qu'une cohésion sociale durable. Néanmoins, on entend souvent parler des nombreux problèmes spécifiques que rencontrent les institutions publiques. L'accumulation de ces problèmes, qui souvent sont d'ordre technique ou logistique, mène à une baisse de productivité et à des pertes économiques qui peuvent être assez conséquentes.

Afin de permettre aux organisations publiques de mieux gérer leur organisation interne et de créer un meilleur service pour les citoyens, la société Noviat a développé l'ARP (Administration Ressource Planning), un ERP (Enterprise Resource Planning) spécialement conçu pour le secteur public sur base du progiciel Odoo. En effet Odoo permet l'utilisation d'un seul environnement centralisé ce qui élimine les communications croisées ainsi que les problèmes de réplication de données.

L'objectif de ce travail consiste d'une part, à développer un module Odoo des à la gestion des sanctions administratives communales (SAC) et qui sera implémenté dans l'ARP, et de l'autre, à développer une application mobile Android qui permettra aux constatateurs d'infractions d'encoder des nouvelles sanctions et qui devra être capable de communiquer avec le module en question.

CAHIER DES CHARGES RELATIF
au TRAVAIL DE FIN D'ETUDES de
EL BAKKOURY Yassine inscrit en 5MIN.

Année académique : 2020-2021

Titre provisoire : Développement d'un module Odoo pour la gestion des sanctions administratives communales (SAC)

Objectifs à atteindre :

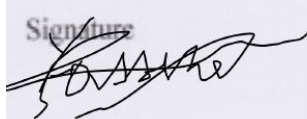

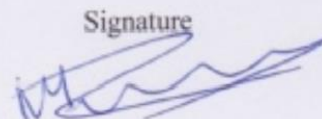
Le module devra idéalement :

- Être actif dans l'ensemble des matières traitées par la problématique des SAC
- Faciliter dans toute la mesure du possible les étapes et les procédures de l'établissement des SAC dans le respect des délais imposés
- Permettre l'archivage des données anciennes
- Permettre à tous les acteurs de réaliser les tâches qui leur incombent dans le respect absolu de la sécurité, confidentialité et du respect de la vie privée
- Permettre l'édition de statistiques en fonction du temps et de la géographie et selon les catégorisations
- Pouvoir générer un dossier qui contienne toutes les pièces, leurs origines et les dates qui y sont attachées.
- Permettre autant que possible l'interaction avec des systèmes électroniques ou informatiques périphériques tant pour l'acquisition que pour l'exportation des données et des transactions financières
- Permettre l'interaction avec l'huissier

Principales étapes :

- Analyse de l'existant
- Collection des besoins et cas d'utilisations et définitions des processus : liste exhaustive et catégorisation des matières traitées par la problématique des SAC et visualisation de l'interface du module + catégorisation des tâches des différents acteurs
- Générer et permettre le suivi d'une SAC
- Gérer les affichages selon les catégories
- Intégration de l'interaction avec des systèmes périphériques (intégration dans l'ARP)

Fait en trois exemplaires à Bruxelles, le 03/12/2020

L'Etudiant	Le Tuteur	Le Promoteur
EL BAKKOURY Yassine	TAYMANS Rémy	PIRENNE Marc
	Département/Unité	Société
	Informatique	Noviat SA
Signature	Signature	Signature
		

Impact du COVID-19 sur la réalisation du TFE :

Suite aux conditions particulières liées aux mesures du COVID-19, l'ensemble du projet a dû être réalisé à distance et ce durant toute la durée du travail de fin d'étude. Etant donné la difficulté d'atteindre une grande partie des objectifs énoncés dans le cahier de charges, qui requièrent notamment la présence d'une personne compétente pouvant m'aider à mieux comprendre le fonctionnement technique de l'ARP, l'ensemble des objectifs de départ ont été revu avec mon promoteur.

Ainsi, nous avons réduit en grande partie les objectifs fixés pour le développement du module Odoo, et nous les avons substitués par le développement d'une application mobile ayant pour but de permettre aux agents constatateurs d'infractions de générer des SAC et de les envoyer vers le serveur Odoo. Les données de l'application mobile devront également être synchronisées avec celles du serveur Odoo.

Table des matières

Impact du COVID-19 sur la réalisation du TFE :	6
I. Introduction	9
II. Contexte	10
1. Description de la société	10
2. Odoo	10
3. ARP	10
III. Analyse du problème	11
1. Les sanctions administratives communales	11
2. Analyse de l'existant	11
3. Etude des cas d'utilisation	12
IV. Description générale du projet	14
1. Fonctionnement général de l'application Flutter	14
2. Fonctionnement général du module Odoo	15
3. Synchronisation de l'application avec le serveur Odoo	15
Du serveur Odoo vers l'application mobile	15
De l'application mobile vers le serveur Odoo	16
V. Application Flutter	17
1. Objectifs de l'application	17
2. Le framework Flutter	17
Popularité	18
Avantages de Flutter	18
3. Structure de la base de données de l'application Flutter	19
Description des tables et des champs	19
4. Diagramme de classe	20
5. Structure du projet	20
Librairies utilisées	21
6. Description des différentes fonctionnalités	22
Authentification de l'utilisateur	22
Affichage de la liste des SAC	22
Création d'une nouvelle SAC	23
Modification/suppression d'une SAC	23
Affichage de la liste des SAC d'un contrevenant	24

VI.	Module Odoo	25
1.	Aspects techniques d'Odoo	25
	L'architecture MVC	25
	Structure d'un module Odoo	26
	L'héritage dans Odoo	26
2.	Structure de la base de données du module	28
	Description des tables et des champs	28
3.	Diagramme de classe	30
4.	Conception du module	30
	Les modèles	30
	Les vues	30
	Le contrôleur	31
5.	Gestion de la sécurité	32
VII.	Synchronisation Flutter-Odoo	33
1.	Synchronisation des données de connexions	33
2.	Synchronisation des SAC avec le serveur Odoo	34
3.	Synchronisation des données des contrevenants et des types de SAC	34
4.	Structure des requêtes http	35
	Réception des SAC du constatateur et des données des contrevenants	35
	Authentification du constatateur	35
	Envoi des nouveaux contrevenants	35
	Envoi des nouvelles SAC vers le serveur Odoo	36
	Envoi des images	36
VIII.	Conclusion et perspectives	37
1.	Application Flutter	37
2.	Module Odoo	37
3.	Synchronisation Flutter-Odoo	37
IX.	Bilan personnel	39
X.	Bibliographie	40
XI.	Annexes	41

I. Introduction

Ce travail de fin d'études a pour objectif le développement d'une solution destinée aux administrations communales, permettant de faciliter leur gestion des SAC (sanctions administratives communales), et s'intégrant à leur structure déjà existante. Ce projet a été réalisé en télétravail au sein de la société Noviat SA, spécialisée dans l'intégration et le développement de solutions basées sur l'ERP Odoo.

Concrètement, ce travail est constitué de trois parties :

- Le développement d'une application mobile réalisée avec le framework Flutter
- Le développement d'un module Odoo
- La connexion entre l'application Flutter et le module Odoo

Cet écrit débutera avec une présentation détaillée du contexte dans lequel s'inscrit le travail, suivi d'une analyse approfondie de la problématique et d'un état de l'art des solutions existantes actuellement. Le chapitre suivant contiendra une description générale du projet, afin d'avoir une vision globale des différentes parties dont les développements techniques seront abordés dans les chapitres suivants.

II. Contexte

1. Description de la société

Noviat SA est une entreprise située à Bruxelles, fondée en 2009 par Mark Pirenne et Luc de Meyer. C'est une entreprise partenaire de Odoo SA, une autre entreprise belge qui développe un ERP (Enterprise Resource Planning) open-source comprenant de très nombreux modules permettant de répondre à de nombreux besoins de gestion des entreprises.

La société est spécialisée dans l'intégration et le développement sur mesure de solutions basées sur l'ERP Odoo dans les entreprises. Elle est également membre de l'OCA (Odoo Community Association), une organisation qui promeut l'utilisation généralisée d'Odoo et soutient le développement collaboratif des fonctionnalités d'Odoo. Le fait d'être membre permet également à Noviat d'acquérir une certaine renommée en tant qu'intégrateur Odoo.

2. Odoo

Odoo est un logiciel ERP open-source développé par l'entreprise belge portant le même nom (Odoo SA) fondée par Fabien Pinckaers. Cet ERP est composé de plusieurs modules ayant pour but de simplifier la gestion des entreprises en leur permettant d'utiliser un seul environnement centralisé ce qui élimine les communications croisées ainsi que les problèmes de réplication de données. L'utilisation de l'ERP permet également de faciliter la circulation de l'information entre toutes les fonctions de l'entreprise et de gérer les contacts avec les parties prenantes externes.

Le logiciel Odoo existe en deux versions : Odoo Community qui est un logiciel gratuit contenant les fonctionnalités de base d'Odoo et Odoo Enterprise qui est payant et apporte des fonctionnalités supplémentaires à Odoo Community. Afin de s'adapter aux différents processus de productions des entreprises, Odoo SA fournit également un service de personnalisation pour Odoo Enterprise.

3. ARP

De manière générale, Noviat aide des entreprises du secteurs privées en intégrant des solutions pour différents modules d'Odoo, ou en adaptant certaines fonctionnalités au besoin des entreprises. Cependant, Noviat travaille également avec des administrations du secteur public comme la Ville de Liège, la Commune d'Ixelles et la Commune de Saint-Gilles qui souhaitent intégrer l'ERP dans leur administration. Pour ce dernier type de client, Noviat a développé l'ARP (Administration Ressource Planning), une déclinaison d'Odoo conçue spécialement pour le secteur public.

L'ARP permet donc aux administrations publiques de structurer les différents domaines fonctionnels, de faciliter l'accès à une information centralisée et pertinente pour l'ensemble de leurs agents, et aussi d'améliorer leurs services rendus aux citoyens.

III. Analyse du problème

1. Les sanctions administratives communales

Le système des sanctions administratives communales (SAC), est un système existant depuis 1999 permettant aux communes d'intervenir plus rapidement et plus efficacement contre les nuisances publiques. Ce système leur permet également de collaborer avec les zones de police dans la lutte contre les incivilités ayant un impact négatif sur la qualité de vie des citoyens, en reprenant les infractions punissables de SAC dans leur règlement de police.

Les incivilités punissables de SAC peuvent être constatées par la police locale ou par d'autres personnes, comme les agents communaux et régionaux, et les gardiens de la paix. L'amende administrative est infligée par le fonctionnaires sanctionnateur, qui est désigné par le conseil communal.

Procédure administrative :

Les comportements pouvant sanctionnés par une amende administrative sont distingués en trois catégories :

- Les nuisances commises par des personnes
- Les nuisances commises par des établissements
- Les infractions en matière d'arrêt et stationnement

Le constat d'infraction doit être transmis au fonctionnaire sanctionnateur dans les deux mois suivant la constatation de l'infraction.

Le fonctionnaire sanctionnateur envoie une copie du constat au contrevenant en lui communiquant les faits de l'infraction.

Le contrevenant a ensuite 15 jours pour se défendre et demander à être entendu si le montant de l'amende prévue est supérieur à 70 euros. Après expiration de ce délai, le fonctionnaire sanctionnateur peut infliger l'amende. Le contrevenant doit alors payer l'amende ou introduire une plainte auprès du tribunal de police dans un délai d'un mois.

2. Analyse de l'existant

Actuellement, la plupart des communes de la région de Bruxelles capitale utilisent une application développée par la société Inforius qui leur permet de faciliter grandement leur gestion des SAC.

Cette application permet :

- **Aux agents constatateurs**, d'encoder les données relatives au contrevenant et à l'infraction commise sur un appareil mobile et donc de générer rapidement un constat qui sera ensuite envoyé vers le fonctionnaire sanctionnateur,
- **Au fonctionnaire sanctionnateur**, d'enregistrer et de suivre des dossiers tout en ayant un aperçu des délais pour chaque dossier et en ayant la possibilité de prendre facilement contact avec les différents acteurs (contrevenant, constatateur,

médiateur, ...). Le fonctionnaire peut également intégrer des modèles de document et générer automatiquement des rapports et des statistiques.

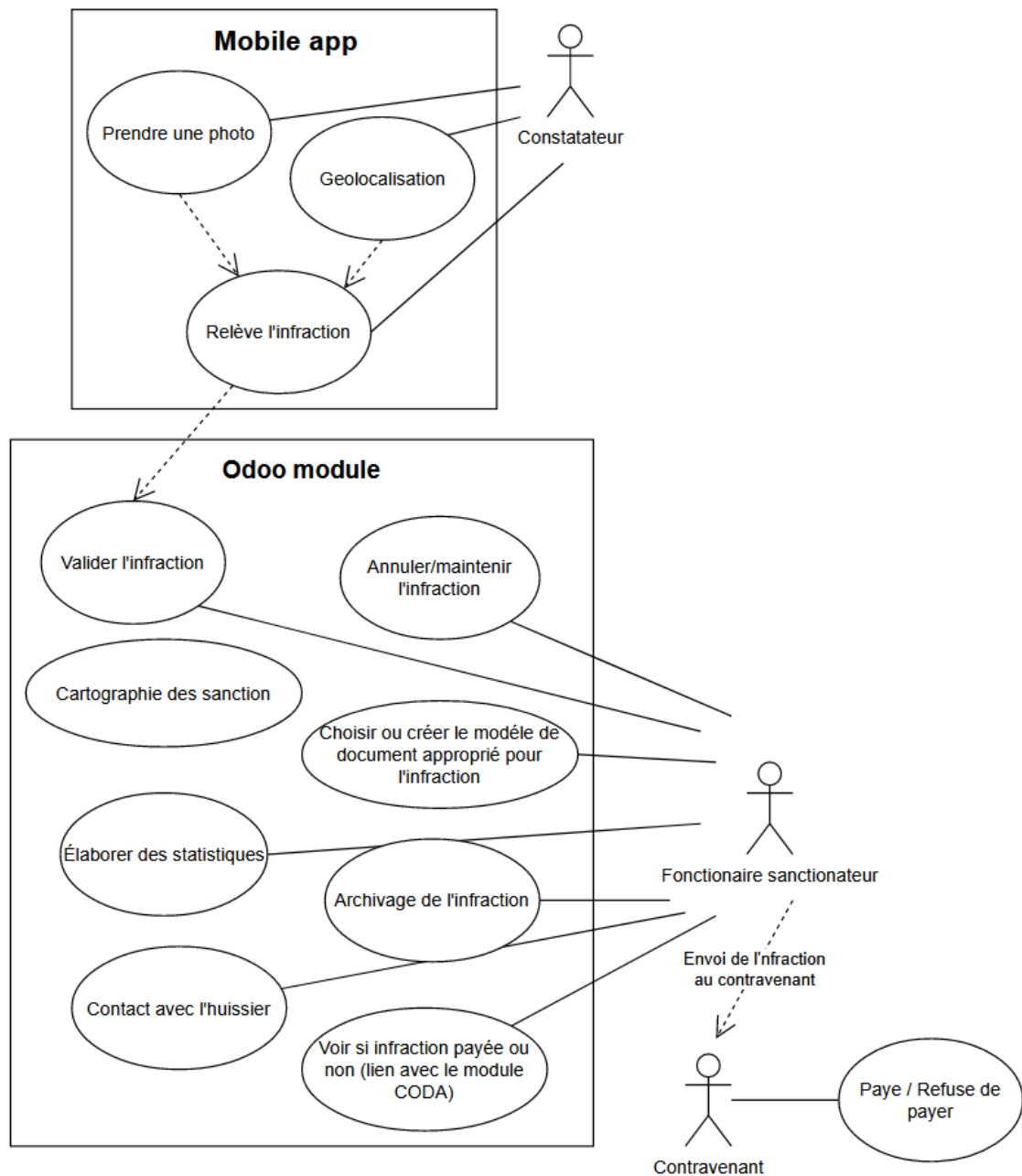
Inforius possède un partenariat avec la société Ipex, une entreprise spécialisée dans les solutions d'impression, ce qui permet à l'application de s'occuper automatiquement de l'impression et de l'envoi du courrier du fonctionnaire sanctionnateur.

3. Etude des cas d'utilisation

Le but de ce travail est donc d'offrir une solution qui permettra à différents acteurs de réaliser des actions similaires à celles qu'ils peuvent effectuer avec l'application d'Inforius tout en étant intégré dans l'ARP.

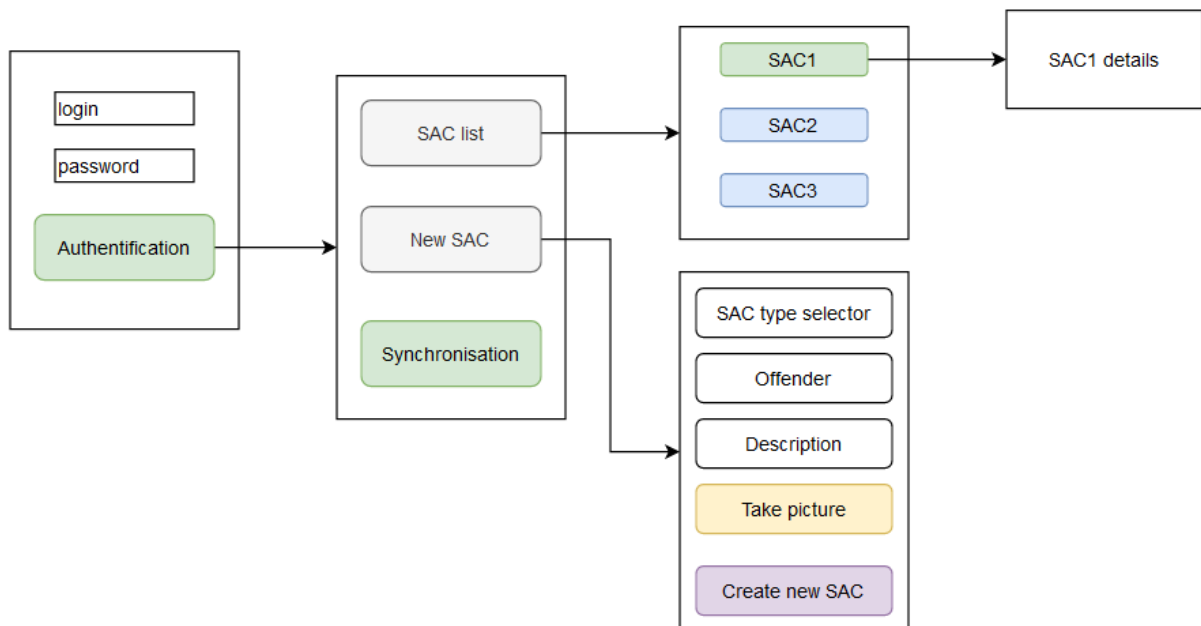
Le travail sera constitué en deux parties :

- **Une application mobile** qui permettra aux constatateurs de relever des infractions tout en leur permettant d'y associer des photos prises sur le terrain. L'application devra également pouvoir utiliser la géolocalisation et fonctionner hors ligne. La base de données de l'application doit également pouvoir se synchroniser avec le serveur Odoo.
- **Un module Odoo** qui permettra au fonctionnaire sanctionnateur de valider ou non les sanctions constatées et de gérer et suivre les dossiers. Il devra également pouvoir prendre contact avec les différents acteurs même avec l'huissier si nécessaire.



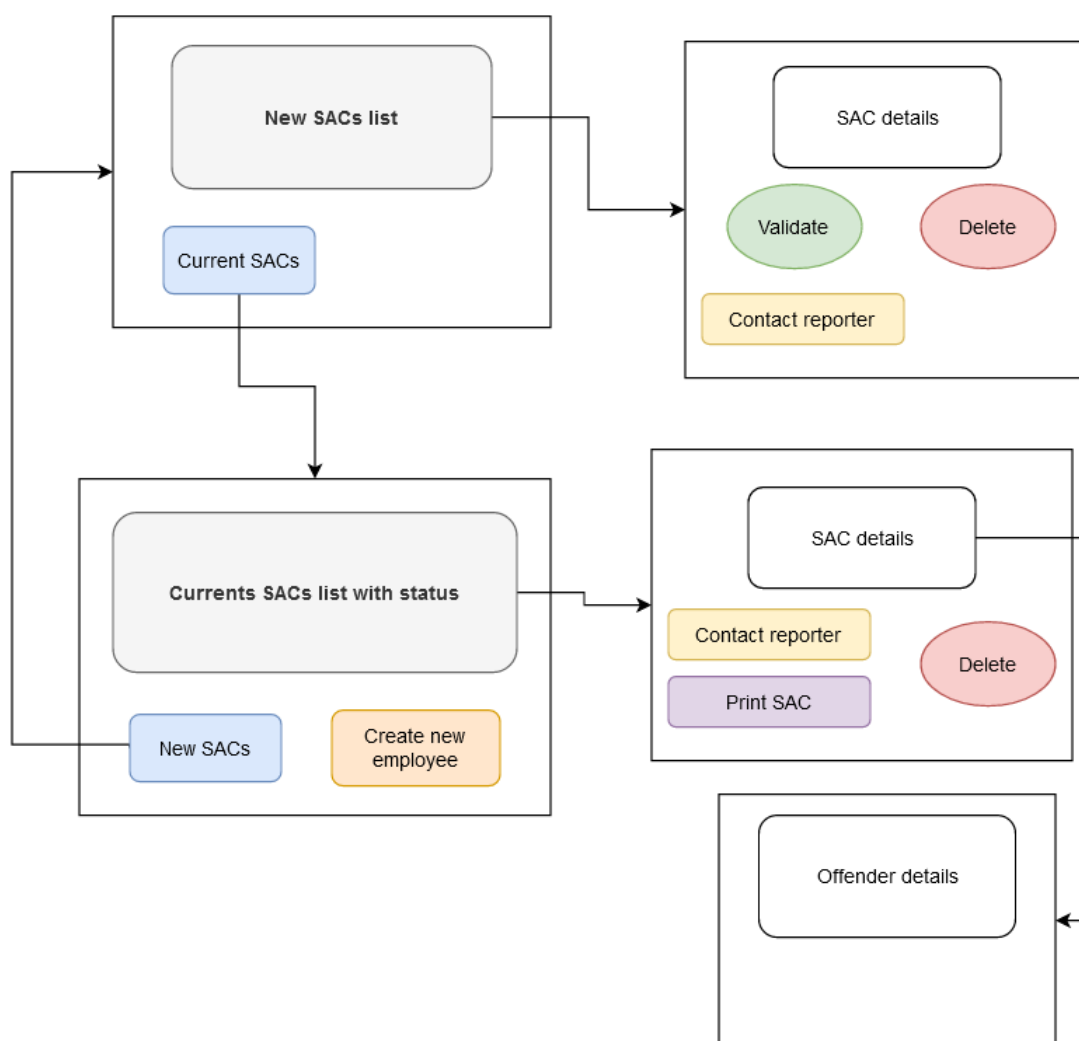
IV. Description générale du projet

1. Fonctionnement général de l'application Flutter



1. L'utilisateur doit tout d'abord s'authentifier à l'aide de son identifiant et son mot de passe avant de pouvoir utiliser l'application.
2. L'utilisateur est ensuite renvoyé vers la page d'accueil où il a la possibilité de visualiser la liste des SAC et voir si elles ont été envoyées vers le serveur Odoo ou non, de créer de nouvelles SAC et de synchroniser les données de l'application avec celles du serveur Odoo.
3. L'utilisateur peut également accéder aux informations et à la liste des sanctions d'un contrevenant en particulier.

2. Fonctionnement général du module Odoo



Après s'être connecter au serveur et au moment où il accède au module, l'utilisateur (l'agent traitant) accède à la liste de toutes les nouvelles SAC qui viennent d'être envoyées depuis l'application Flutter. Il peut ensuite accéder au détail de chaque sanction et décider de valider ou non la SAC. L'utilisateur a également accès à la liste des SAC validées où il peut imprimer une SAC selon un modèle d'impression qu'il aura choisi afin de pouvoir l'envoyer au contrevenant. L'utilisateur peut également prendre contact avec le constatateur d'une SAC et assigner le poste d'agent constatateur à d'autres employés de l'organisation.

3. Synchronisation de l'application avec le serveur Odoo

Du serveur Odoo vers l'application mobile :

Au départ toutes les données concernant les informations de connexion des constatateurs (login et mot de passe) ainsi doivent être copiées dans la base de données de l'application mobile depuis le serveur Odoo.

De l'application mobile vers le serveur Odoo :

Les informations envoyées vers le serveur Odoo depuis l'application concernent toutes les données relatives aux nouvelles sanctions qui viennent d'être constatées ainsi que les informations sur de nouveaux contrevenants.

V. Application Flutter

1. Objectifs de l'application

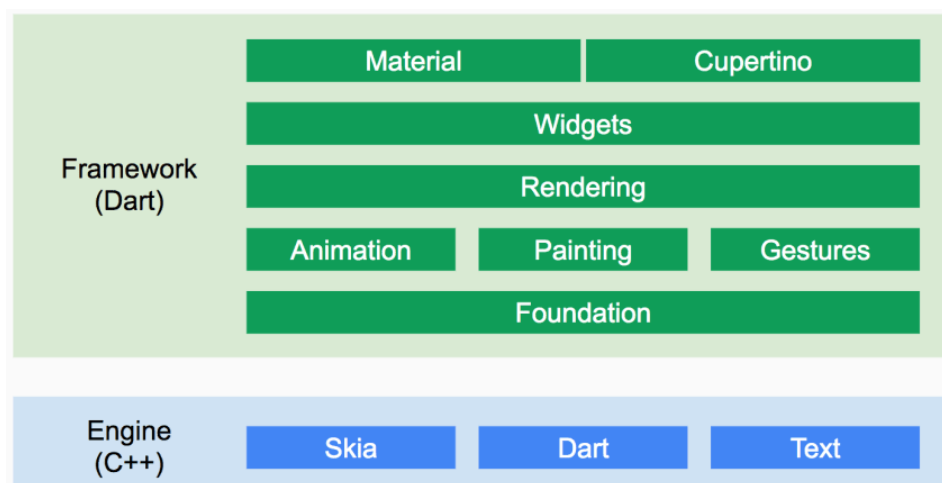
Voici les différents services que devra offrir l'application aux constatateurs d'infractions :

- Authentification de l'utilisateur (le constatateur)
- Rédaction d'un constat d'infractions SAC
- Joindre des photos pour chaque SAC
- Détection automatique du lieu du constat par géolocalisation
- Identification du contrevenant
- Synchronisation des données avec le serveur Odoo
 - Envoi des nouvelles SAC
 - Synchronisation des données d'authentification
 - Synchronisation des données des contrevenants

2. Le framework Flutter

Le développement de l'application se fera avec Flutter, un framework open-source qui existe depuis 2018 et développé par Google. Les applications Flutter sont écrites en Dart, un langage de programmation orienté objet utilisant le moteur graphique Skia C++, également conçu par Google.

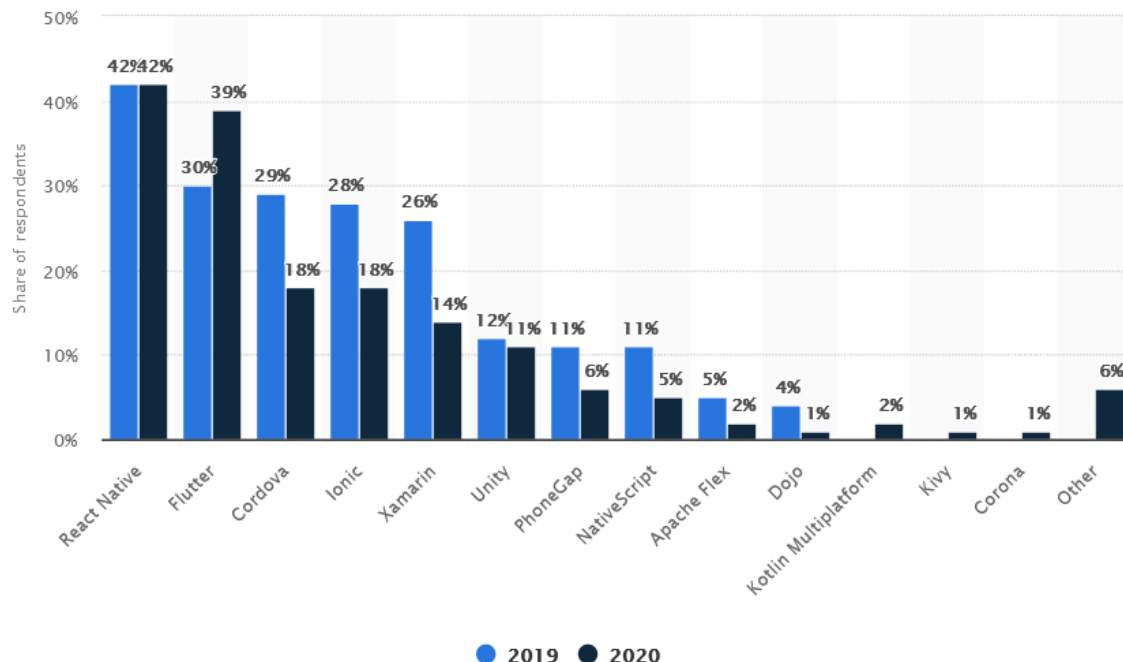
Le principe de Flutter repose sur l'utilisation des widgets et des composants. Les widgets permettent d'utiliser la programmation orienté objet en ayant directement un rendu sur l'interface utilisateur. Concrètement les widgets représentent chaque bouton et texte affiché dans l'interface.



Architecture de Flutter

Popularité :

Selon une enquête réalisée en 2020 par Statista, Flutter est le deuxième framework de développement d'applications multiplateforme le plus utilisé par les développeurs de logiciels dans le monde avec une forte augmentation (9%) d'utilisateurs en 2020.

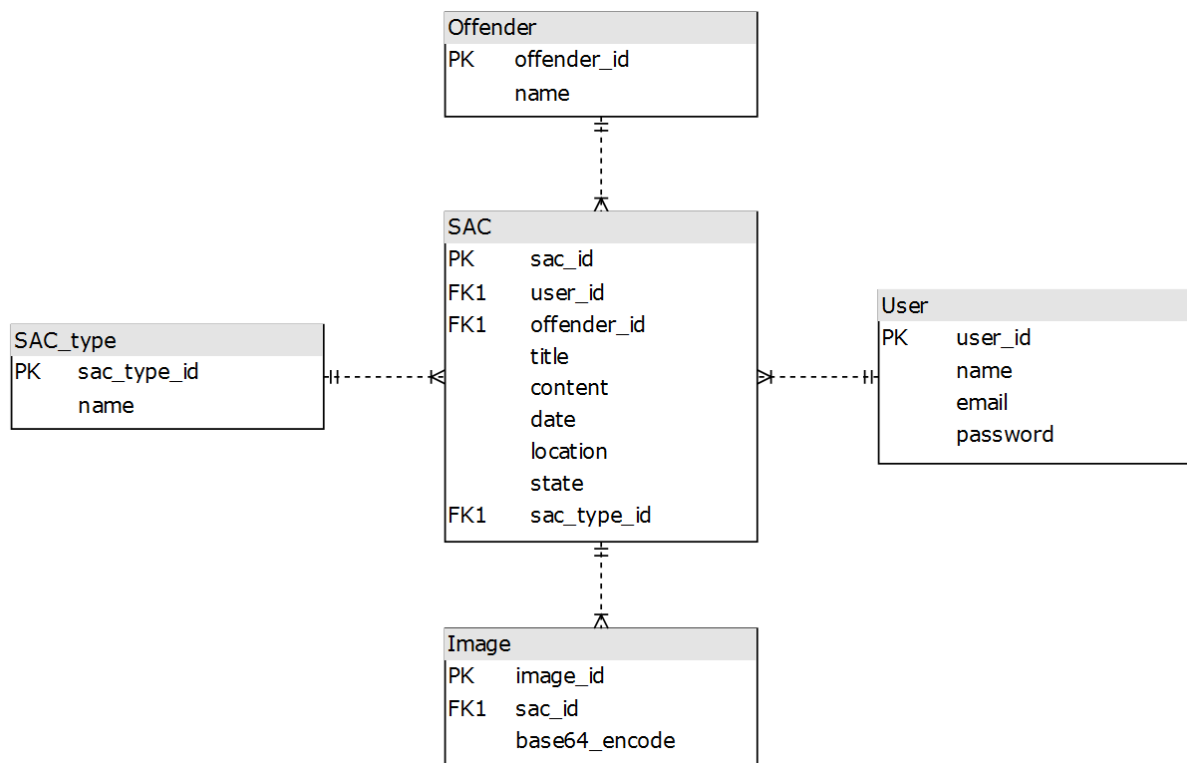


Cross-platform mobile frameworks used by software developers worldwide in 2019 and 2020
(<https://www.statista.com/statistics/869224>)

Avantages de Flutter :

- Flutter est un framework multiplateforme ce qui permet de développer des applications pour Android, iOS, Linux, Mac, Windows, Google Fuchsia et le web à partir d'une seule base de code. On évite ainsi les problèmes de cohérence de l'interface utilisateur sur différentes plates-formes. De plus grâce au moteur Skia, les applications Flutter ne sont pas affectées par une mise à jour ou une personnalisation du système d'exploitation et garderont donc la même apparence.
- Le langage Dart propose deux modes de compilation : JIT (Just In time) et AOT (Ahead Of Time). Avec AOT le code sera optimisé directement pour l'architecture sur laquelle il fonctionnera ce qui permet ainsi d'avoir de meilleures performances et de réduire la taille de l'application. JIT offre la fonctionnalité de Hot Reload lors des compilations, ce qui permet de réduire le temps entre chaque build à quelques millisecondes et facilite ainsi le développement des applications. A titre de comparaison, la compilation d'une application Android de taille moyenne habituelle prend au moins 40 secondes pour être expédiée sur l'appareil de test.

3. Structure de la base de données de l'application Flutter

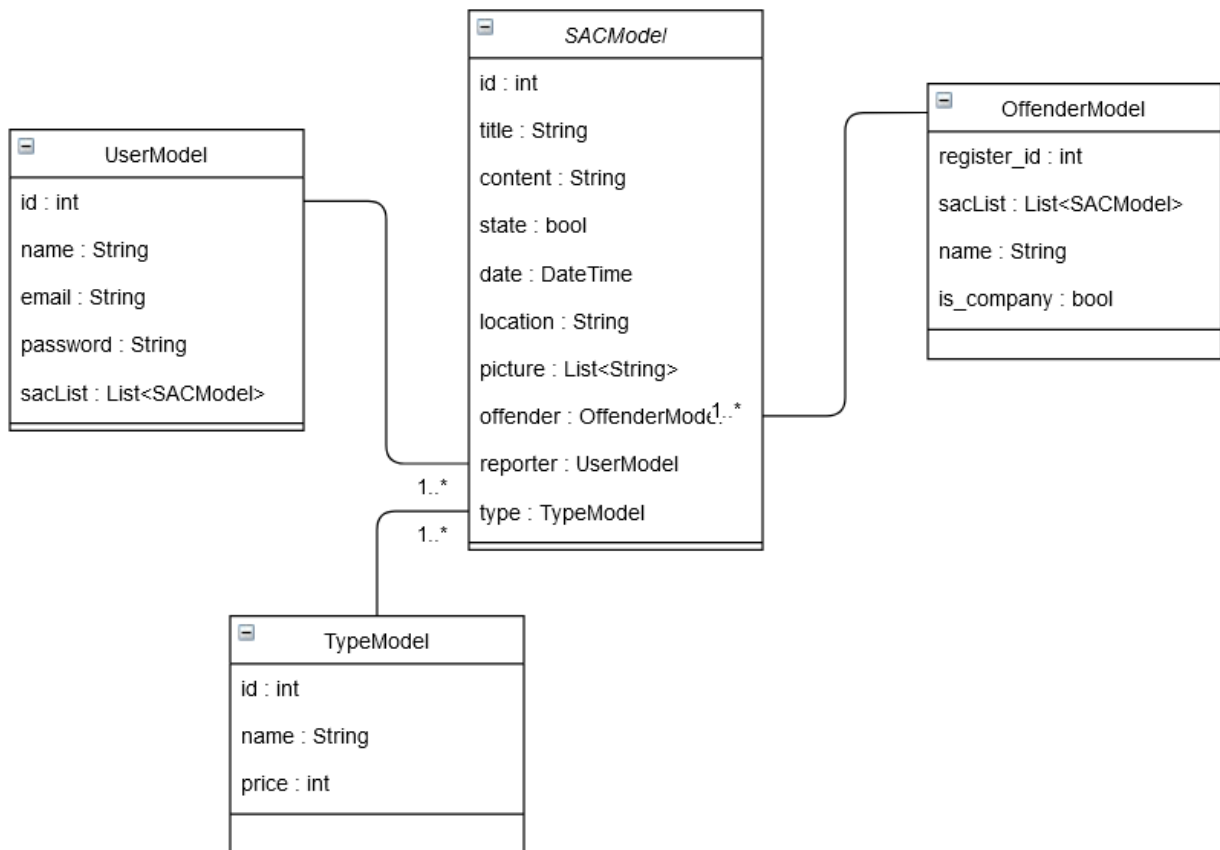


Description des tables et des champs :

- **SAC** : représente un constat d'infraction SAC
 - user_id : id de l'utilisateur de l'application ayant constaté l'infraction
 - offender_id : id du contrevenant
 - title : titre de la sanction
 - content : description de l'infraction
 - date : date du constat d'infraction
 - location : lieu du constat d'infraction
 - state : état de la sanction (envoyée vers le serveur Odoo ou non)
 - sac_type_id : id du type de sanction
- **Image** : représente une image associée à une SAC
 - sac_id : id de la SAC associée à l'image
 - base64code : image en format base64
- **SAC_type** : représente un type de SAC
 - name : nom du type
 - price : tarif de base de l'amende SAC

- **Offender** : représente un contrevenant
 - `is_company` : indique si le contrevenant est une entreprise (valeur = False s'il s'agit d'une personne)
 - `national_register` : numéro de registre national du contrevenant (ou numéro d'entreprise si `is_company='True'`)
 - `name` : nom du contrevenant
- **User** : représente l'utilisateur de l'application (le constatataire)
 - `name` : nom de l'utilisateur
 - `email` : email de l'utilisateur (pour l'authentification)
 - `password` : mot de passe de l'utilisateur (pour l'authentification)

4. Diagramme de classe



5. Structure du projet

- *Components*
 - **Cards.dart** : définit un composant SAC lors de l'affichage de la liste des SAC
 - **Faderoute.dart** : définit le thème (couleur) de l'application

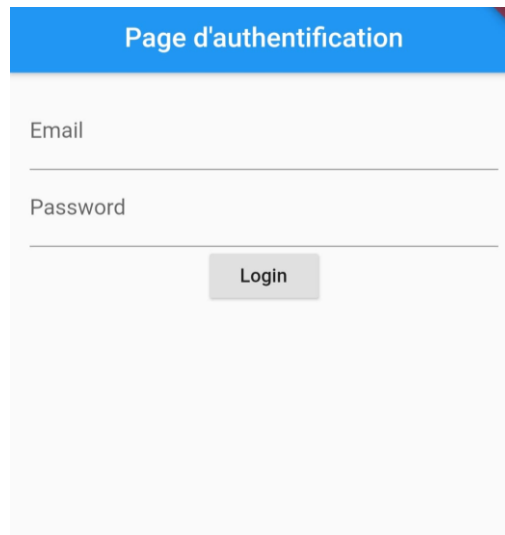
- *Screens*
 - **edit.dart** : page de création/modification de SAC
 - **home.dart** : page d'accueil, affiche la liste de toutes les SAC
 - **offender.dart** : affichage de la liste des sanctions d'un contrevenant en particulier
 - **View.dart** : affichage des informations concernant une SAC
- *Services*
 - **database.dart** : contient les fonctions d'interaction avec la base de données
 - **odoo.dart** : contient les fonctions d'interaction avec le serveur Odoo
- **main.dart** : page d'authentification de l'utilisateur
- **pubspec.yaml** : définit l'ensemble des librairies utilisées

Librairies utilisées :

- **sqflite** : permet d'effectuer des transactions avec une base de donnée SQLite.
- **intl** : fournit des fonctions d'internationalisation et de localisation, notamment la traduction de messages, les pluriels et les genres, le formatage et l'analyse syntaxique des dates et des nombres, et le texte bidirectionnel.
- **location** : permet d'obtenir les coordonnées géographiques par géolocalisation
- **geocoder** : permet de trouver une adresse à partir des coordonnées géographiques
- **camera** : permet à l'application d'accéder à l'appareil photo
- **path_provider** : permet à l'application d'accéder au système de fichiers
- **image_picker** : permet de choisir des images dans la galerie et de prendre de nouvelles photos avec l'appareil photo.
- **http** : permet de faire des requêtes http

6. Description des différentes fonctionnalités

Authentification de l'utilisateur :



Page d'authentification

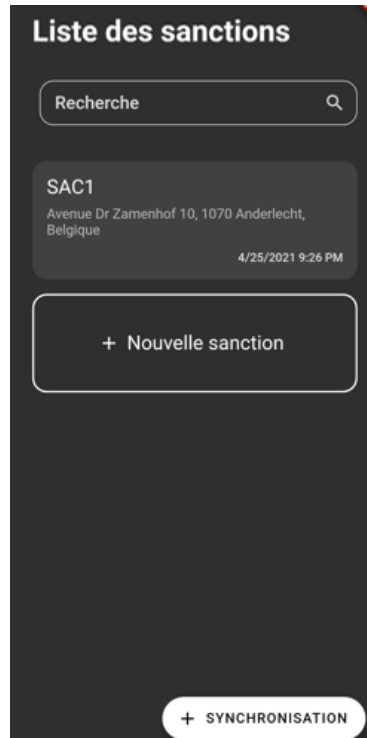
Email

Password

Login

Lorsqu'il ouvre l'application, l'utilisateur doit tout d'abord s'authentifier à l'aide son identifiant et de son mot de passe. La première fois qu'il se connecte, l'application doit être connectée au serveur afin de copier les données du serveur Odoo de tous les constatateurs dans la base de données Flutter.

Affichage de la liste des SAC :



Liste des sanctions

Recherche

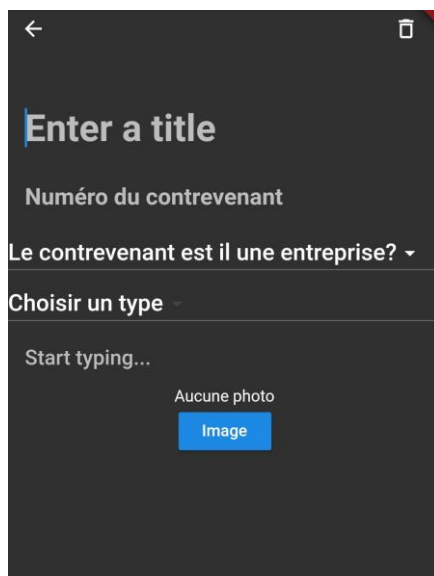
SAC1
Avenue Dr Zamenhof 10, 1070 Anderlecht,
Belgique
4/25/2021 9:26 PM

+ Nouvelle sanction

+ SYNCHRONISATION

Après l'authentification, l'utilisateur est dirigé vers la page d'accueil qui affiche la liste de toutes les SAC qui se trouvent dans la base de données de l'application.

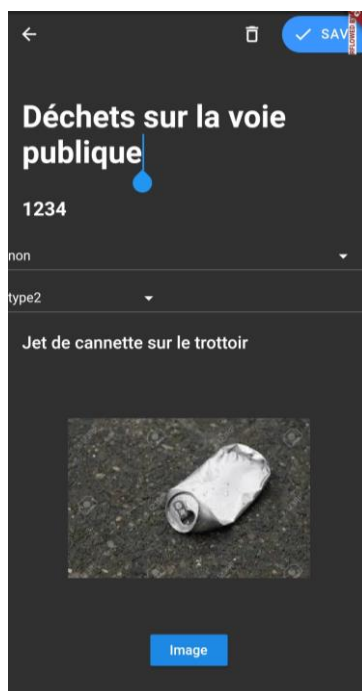
Création d'une nouvelle SAC :

A screenshot of a mobile application form for creating a new SAC. The form has a dark background with white text. At the top, there is a back arrow and a trash icon. The main title is "Enter a title". Below it is a field for "Numéro du contrevenant". Then a dropdown menu labeled "Le contrevenant est il une entreprise? ▾". Below that is another dropdown menu labeled "Choisir un type ▾". At the bottom, there is a text input field with the placeholder "Start typing...". Below the input field is a section labeled "Aucune photo" with a blue button labeled "Image".

Lorsque l'utilisateur constate une nouvelle infraction, il peut créer une nouvelle sanction en indiquant dans le formulaire si le contrevenant est une entreprise ou une personne ainsi que son numéro d'identification, le type de sanction, ainsi qu'une description de l'infraction. Il peut également y joindre une ou plusieurs photos.

Au moment où la sanction est enregistrée, la date et l'heure de création sont également enregistrées et le lieu où l'infraction a été constaté est détecté par géolocalisation. On vérifie également si le contrevenant n'est pas déjà répertorié dans la base de données, et si ce n'est pas le cas, un nouvel élément est ajouté dans la table Offender.

Modification/suppression d'une SAC :

A screenshot of a mobile application form for modifying or deleting a SAC. The form has a dark background with white text. At the top, there is a back arrow, a trash icon, and a blue button labeled "SAVE". The main title is "Déchets sur la voie publique". Below it is a field with the value "1234". Then a dropdown menu labeled "non". Below that is another dropdown menu labeled "type2". Below the input field is a section labeled "Jet de cannette sur le trottoir". Below the text is a photo of a crushed aluminum can on a sidewalk. Below the photo is a blue button labeled "Image".

Une fois la SAC créée, il est toujours possible de la modifier ou de la supprimer tant qu'elle n'a pas encore été envoyée vers le serveur Odoo. On ne peut cependant pas modifier le lieu et la date du constat de l'infraction.

Affichage des détails d'une SAC :



Affichage de la liste des SAC d'un contrevenant :



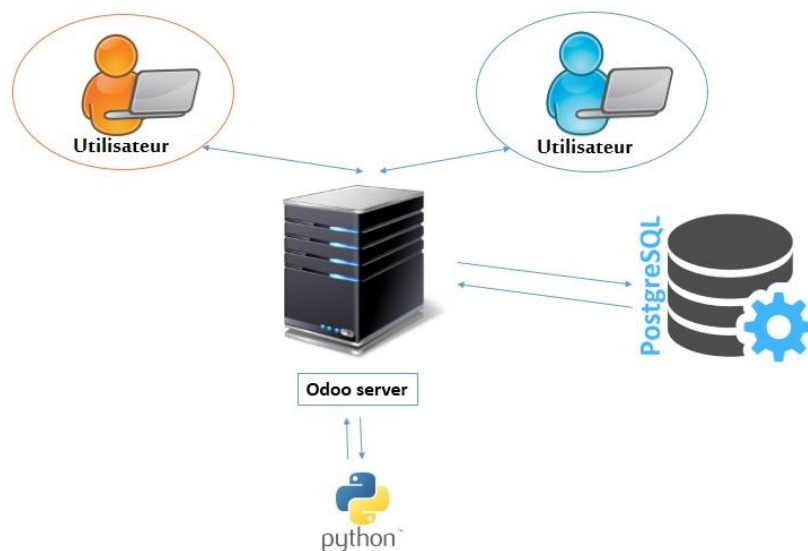
VI. Module Odoo

1. Aspects techniques d'Odoo

Odoo est un logiciel basé sur une architecture client-serveur dans laquelle les clients sont des navigateurs web accédant au serveur Odoo à l'aide du protocole JSON-RPC (Remote Procedure Call).

L'architecture d'Odoo se compose de deux tiers :

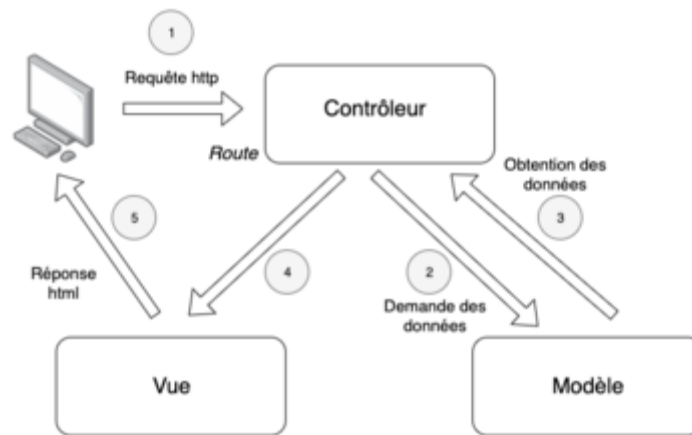
- **Le serveur de base de données PostgreSQL** qui gère l'ensemble bases de données Odoo qui contiennent toutes les données des applications, et aussi la plupart des éléments de configuration du système Odoo.
- **Le serveur Odoo** qui contient toute la logique d'entreprise et assure le fonctionnement d'Odoo. Ce serveur fournit également un ORM (Object-Relational Mapping) qui s'occupe de faire la liaison entre les objets Python et la base de données PostgreSQL tout en gérant les relations entre objet. Le serveur Odoo utilise Werkzeug comme serveur WSGI (Web Server Gateway Interface) qui permet la communication entre un serveur web et une application python.



<https://apcpedaqogie.com/connaitre-larch>

L'architecture MVC :

Le fonctionnement d'Odoo se base sur le patron de conception MVC qui permet de séparer les données et l'interface utilisateur en introduisant un composant intermédiaire : le contrôleur.



<https://fr.wikipedia.org/wiki/Mod%C3%A8le>

Structure d'un module Odoo :

L'architecture technique d'un module Odoo se compose d'un répertoire principal, contenant plusieurs sous-répertoires et des fichiers avec une convention bien définie. Parmi ces fichiers on retrouve :

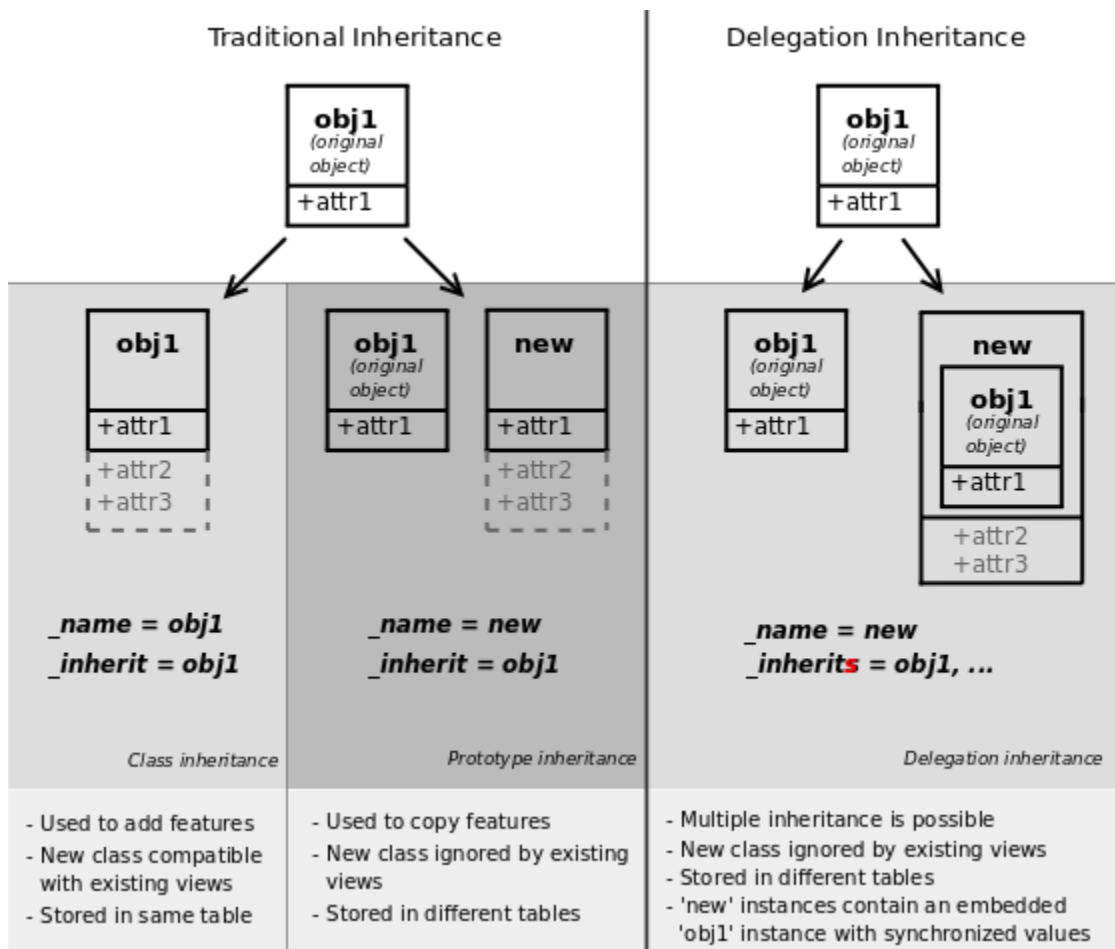
- **__init__.py** : contient les informations d'importation pour les fichiers python du module
- **__manifest__.py** : permet de déclarer un projet python en tant que module Odoo et à spécifier les métadonnées du module
- **models** : contient les classes python liées au différentes tables du schéma de base de données
- **views** : contient les différentes vues du module
- **controller** : contient les contrôleurs web qui prennent en charges les requêtes http provenant du navigateur internet
- **data** : contient les données préchargées au moment de l'installation du module
- **demo** : contient les données de démonstration (test)
- **security** : contient les règles de sécurité
- **static** : contient les fichiers JavaScript

L'héritage dans Odoo :

Odoo permet d'étendre un module existant de manière modulaire grâce à trois mécanismes d'héritage différents :

- **L'héritage classique** : on crée un nouveau modèle à partir d'un modèle déjà existant, les changements apportés aux nouveaux modèles ne modifient pas le modèle de référence.

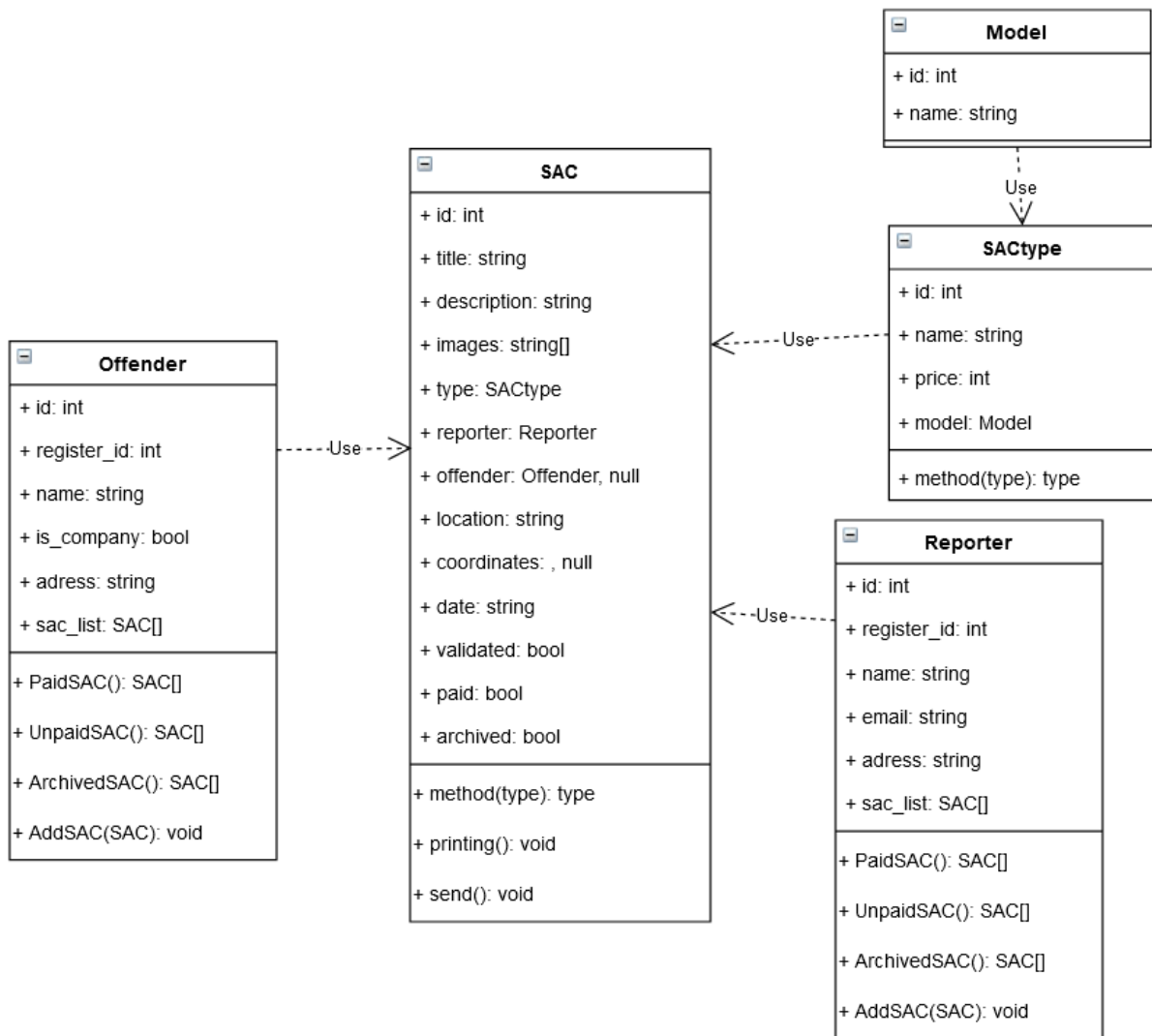
- **L'héritage par extension** : on rajoute des nouveaux champs à un modèle déjà existant.
- **L'héritage par délégation** : ce type d'héritage permet de créer un nouveau modèle héritant de plusieurs modèles déjà existant à la fois.



<https://www.odoo.com/documentation/9.0/howtos/backend.html#inheritance>

- **hr.employee** : représente un employé
 - name : nom de l'employé
 - adress_home_id
 - identification_id
 - user_id : id du compte utilisateur associé à l'employé
- **hr.job** : représente un métier (constatateur)
 - name : nom du métier
- **hr.employee.job** : table de liaison entre job et hr.employee (relation Many to Many), représente un employé affecté au poste d'agent constatateur
 - job_id : id du métier
 - hr.employee_id : id de l'employé
- **sac** : représente un constat d'infraction SAC
 - employee_id : id de l'utilisateur de l'application ayant constaté l'infraction
 - res.partner_id : id du contrevenant
 - title : titre de la sanction
 - content : description de l'infraction
 - date : date du constat d'infraction
 - location : lieu du constat d'infraction
 - state : état de la sanction (envoyée vers le serveur Odoo ou non)
 - sac_type_id : id du type de sanction
- **sac_type** : représente un type de SAC
 - name : nom du type
 - price : tarif de base de l'amende SAC
- **doc_model** : représente un modèle document
 - name : nom du modèle
- **ir_attachment** : représente un lien vers les images de la SAC (ou autres pièces jointes)
 - name : nom de la pièce jointe
 - res_id : id de la SAC
 - res_model : nom du modèle (=sac_module.sac)
 - local_url : lien vers la pièce jointe

3. Diagramme de classe



4. Conception du module

Les modèles :

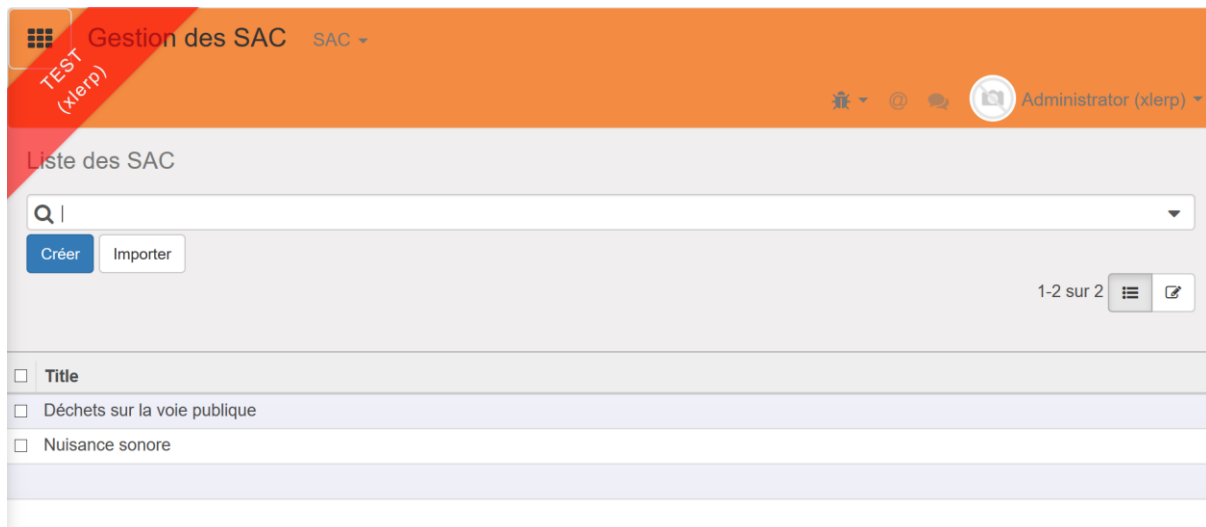
Trois modèles ont été définis au sein du module, chacun lié à une table de la base de données grâce à l'ORM :

- **sac** : modèle qui représente une SAC
- **type_sac** : représente un type de SAC
- **offender** : héritage par extension du modèle res.partner et représente un contrevenant.

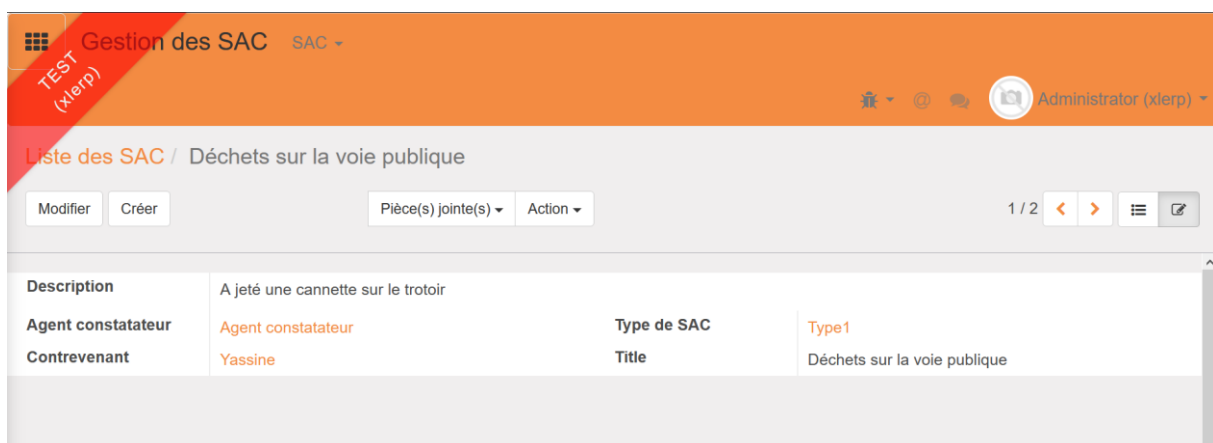
Les vues :

Dans Odoo, l'affichage est basé sur une arborescence de menus, chaque menu possédant des sous menus menant à des actions qui mènent vers des vues. Une vue définit la manière dont les objets d'un modèle sont affichés.

- **Liste des SAC** : on utilise une 'tree view' qui permet d'afficher les éléments d'un modèle sous forme de table ainsi qu'une 'search view' qui permet de filtrer les objets du modèle



- **Détails d'une SAC** : on utilise une 'form view' qui permet d'afficher les données d'un seul objet du modèle



Le contrôleur :

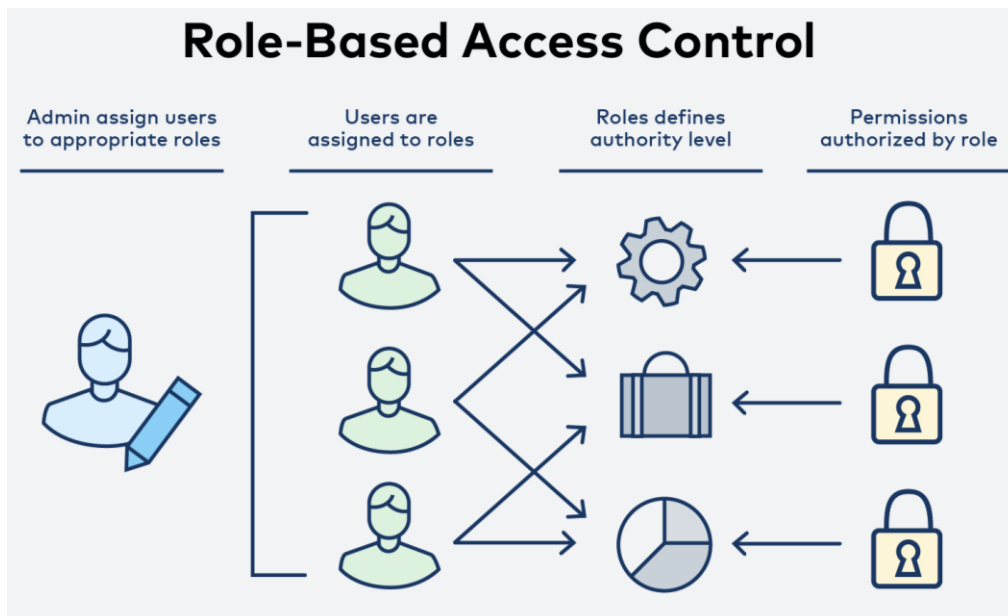
Plusieurs routes ont été définies dans le contrôleur afin de permettre la synchronisation des données avec l'application Flutter (plus de détails au chapitre suivant)

- **/sac_module/new_offender** : permet d'ajouter un nouveau contrevenant
- **/sac_module/new_sac** : permet d'ajouter une nouvelle SAC
- **/sac_module/new_picture** : permet d'ajouter de nouvelles images associées à une SAC
- **/sac_module/get_sac** : renvoie toute les SAC d'un constatateur en particulier
- **/sac_module/get_users** : renvoie les utilisateurs des employés assignés aux poste d'agent constatateur

5. Gestion de la sécurité

Afin de gérer les droits d'accès et de modification dans le module, il est possible avec Odoo de définir des groupes d'accès pour différents types d'utilisateurs, suivant le modèle de sécurité RBAC (Role Based Access Control). Pour cela, on commence par définir les droits d'accès sur les modèles (read, write, create, delete). On définit ensuite des groupes de sécurité liés à plusieurs droits d'accès, chaque groupe pouvant hériter d'un autre (expl : groupe manager hérite du groupe user). Ces groupes peuvent être assignés dans des champs dans des objets, des vues, des actions ou des menus afin de contrôler l'accès à ces données. Enfin on associe les groupes de sécurité aux utilisateurs.

Dans l'ARP, il est possible de définir des user_template, des utilisateurs 'blanc' à qui on lie plusieurs groupes de sécurité. Chaque user_template est lié à un objet hr.job. A chaque connexion d'un utilisateur, on retrouve l'employé qui lui correspond (grâce à LDAP) et on récupère les postes (hr.job) qu'il occupe, puis on associe à l'utilisateur tous les groupes de sécurité liés aux user_template de ces différents postes.



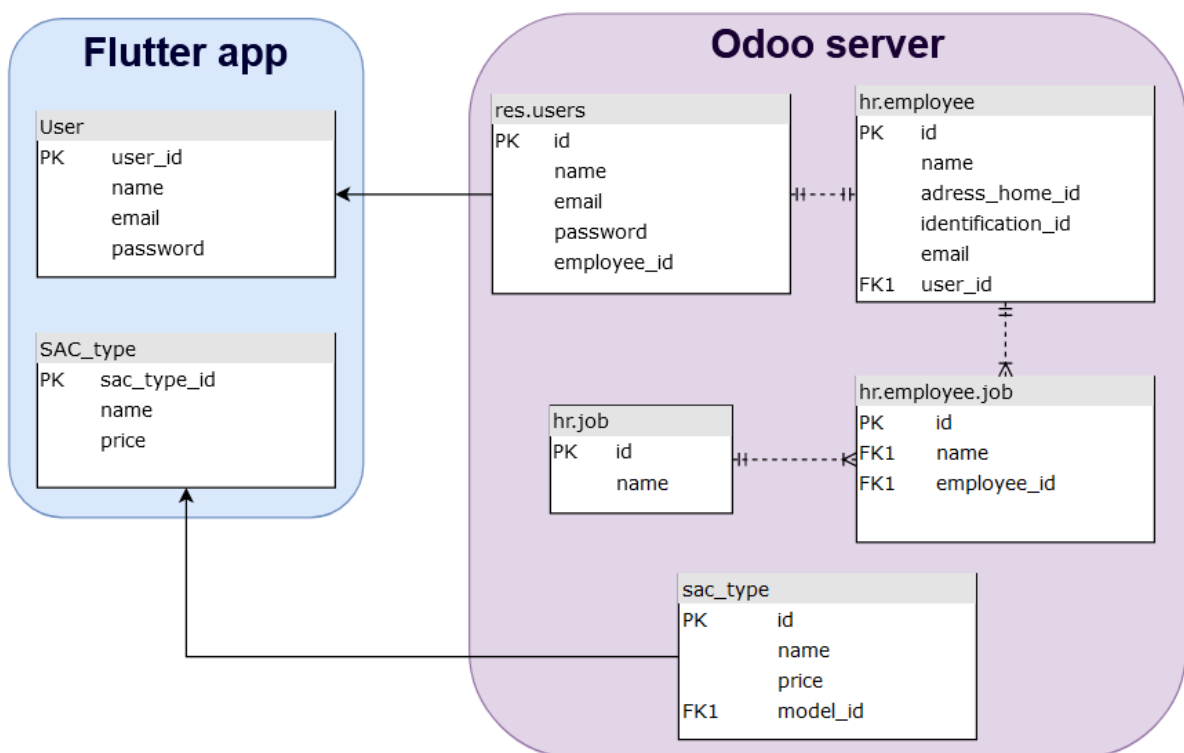
<https://www.chaossearch.io/platform/feat>

Il est également possible de mettre des règles sur des données spécifiques d'un modèle en appliquant des filtres pour certains groupes de sécurité à l'aide des ir_rules.

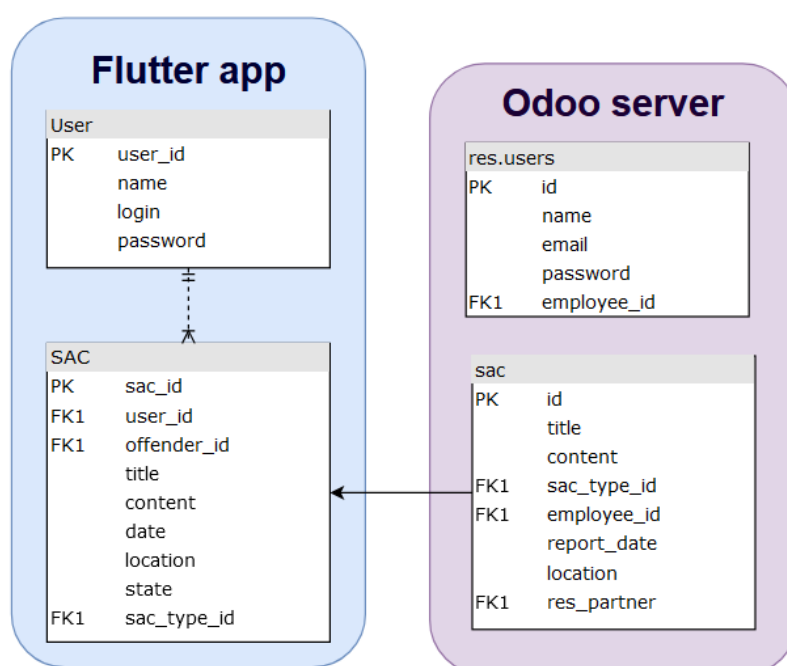
VII. Synchronisation Flutter-Odoo

1. Synchronisation des données de connexions

La première fois que l'on s'authentifie dans l'application Flutter, la base de donnée Flutter copie les données de la table res.users depuis le serveur Odoo dans sa table User. L'application recopie uniquement les données des utilisateurs assigné au poste d'agent constatateur. Pour cela, on commence par chercher l'ensemble des objets contenu dans la table hr.employee.job dont le champ name = 'Agent constatateur', et l'on récupère la valeur du champ employee_id. Enfin, on récupère l'objet de la table res.user dont la valeur du champ employee_id correspond à la valeur récupérée précédemment.

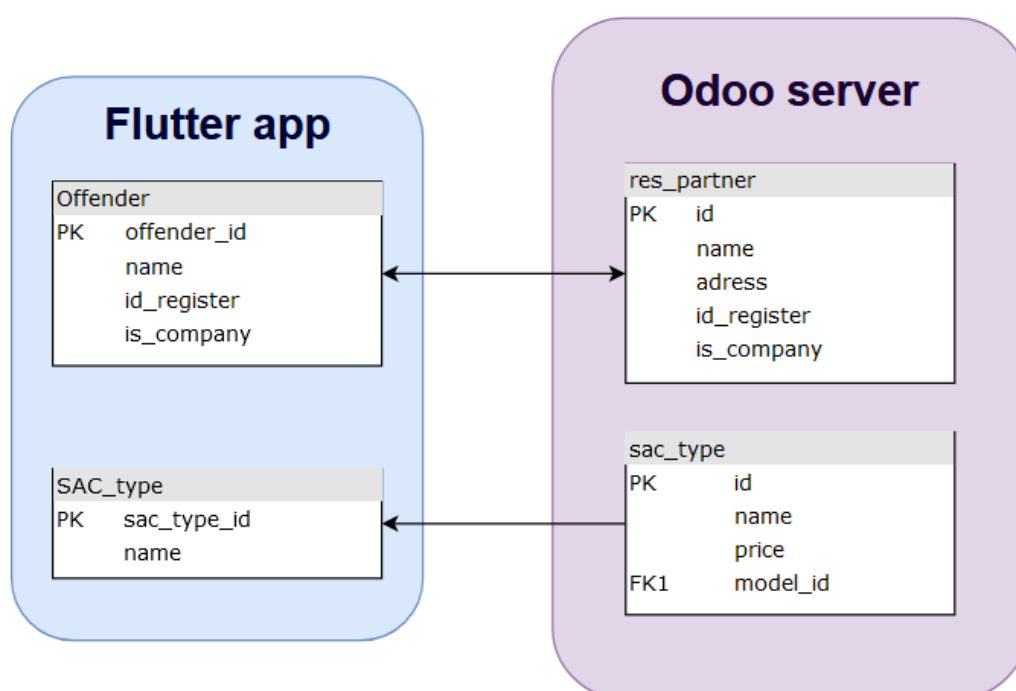


2. Synchronisation des SAC avec le serveur Odoo



Après authentification, l'ensemble des données des SAC du constatateur lié à l'utilisateur, sont copiées dans la base de donnée Flutter depuis le serveur Odoo. Pour cela, on cherche l'ensemble des objets contenus dans la table 'sac' dont le champ 'employee_id' correspond à l'id de l'utilisateur.

3. Synchronisation des données des contrevenants et des types de SAC



Au moment où la base de données Flutter copie les données des SAC du serveur, les données des types de SAC et des contrevenants liés aux SAC, contenus dans les tables `sac_type` et `res_partner`, sont également copiées dans les tables `SAC_type` et `Offender` de Flutter.

Au moment où l'on envoie de nouvelles SAC de l'application Flutter vers le serveur, on vérifie tout d'abord si les informations liées au contrevenant n'existent pas déjà dans la base de données Odoo en comparant le champ 'id_register'.

4. Structure des requêtes http

Réception des SAC du constatateur et des données des contrevenants :

Lorsque le constatateur accède à la page d'accueil, l'application Flutter copie la liste des SAC du constatateur dans sa base de données. Les données des contrevenants liés aux SAC sont également copiées.

GET : http://localhost:8069/sac_module/get_sac

```
{
  "jsonrpc": "2.0",
  "params": {
    "sac_reporter_id": 1
  }
}
```

Authentification du constatateur :

Avant d'envoyer les SAC, l'utilisateur doit d'abord s'authentifier pour prouver qu'il est un utilisateur Odoo et vérifier qu'il est toujours un agent constatateur.

POST : <http://localhost:8069/web/session/authenticate>

```
{
  "jsonrpc": "2.0",
  "params": {
    "db": "xlerp",
    "login": "user_name",
    "password": "password"
  }
}
```

Envoi des nouveaux contrevenants :

Avant d'envoyer les SAC vers le serveur Odoo, il faut d'abord ajouter les données des nouveaux contrevenants.

POST : http://localhost:8069/sac_module/new_offender

```
{
  "jsonrpc": "2.0",
  "params": {
    "offender_name": "Yassine",
    "is_company": false,
    "national_register": "national_register"
  }
}
```

Envoi des nouvelles SAC vers le serveur Odoo:

POST : http://localhost:8069/sac_module/new_sac

```
{
  "jsonrpc": "2.0",
  "params": {
    "sac_id": 1,
    "sac_title": "title",
    "sac_content": "content",
    "sac_offender_national_register": "national_register",
    "sac_reporter_id": 1,
    "sac_date": "01/01/2021",
    "sac_location": "adress",
    "sac_type_id": 1
  }
}
```

Envoi des images :

Une fois les SAC envoyées on envoie les images qui leur sont associées

POST : http://localhost:8069/sac_module/new_picture

```
{
  "jsonrpc": "2.0",
  "params": {
    "picture_id": 1,
    "sac_id": 1,
    "base64code": ""
  }
}
```

VIII. Conclusion et perspectives

Ce dernier chapitre présente le bilan général et perspectives spécifiques à chacune des parties du projet de ce travail de fin d'études.

1. Application Flutter

L'objectif principal de l'application Flutter est de permettre, aux agents constateurs de générer facilement des constats d'infractions, de les visualiser et de les envoyer vers le serveur Odoo. Toutes les fonctionnalités qui ont été prévues pour l'application ont été développées :

- L'utilisateur peut s'authentifier
- L'utilisateur peut créer une SAC
- L'utilisateur peut modifier/supprimer une SAC
- L'utilisateur peut associer des photos à une SAC
- Le lieu du constat d'infraction est détecté par géolocalisation
- L'utilisateur a accès à la liste des SAC d'un contrevenant en particulier

2. Module Odoo

L'objectif du module Odoo est de permettre, aux agents traitants de valider ou non les SAC envoyées par les agents constateurs, et d'avoir un suivi complet de tous les dossiers SAC. Parmi toutes les fonctionnalités qui avaient été prévues, seules quelques-unes ont pu être réalisées :

- L'utilisateur peut synchroniser ses données avec l'application Flutter
- L'utilisateur a accès à la liste des SAC
- L'utilisateur peut afficher les détails d'une SAC
- L'utilisateur (ayant les droits) peut assigner un employé au poste d'agent constatateur

3. Synchronisation Flutter-Odoo

La synchronisation Flutter-Odoo se fait au moment où l'agent constatateur s'authentifie pour la première fois dans l'application mobile, et au moment où ce dernier décide d'envoyer des SAC vers le serveur Odoo. Ce sont les seuls moments où l'application doit être connectée au serveur Odoo.

De Flutter vers Odoo :

- Les SAC peuvent être envoyées vers le serveur Odoo
- Les données des contrevenants sont envoyées avec ceux des SAC

De Odoo vers Flutter :

- L'application récupère les données des types de SAC depuis le serveur Odoo
- L'application récupère les données de connexion des utilisateurs (users des agents constatateurs) depuis le serveur Odoo

Pour ce qui est de l'envoi des informations vers le serveur Odoo, tout fonctionne correctement sauf l'envoi des images attachés au SAC. L'application reçoit correctement les données des types de SAC ainsi que ceux des utilisateurs, mais il reste encore quelques problèmes à résoudre pour la réception des SAC des constatateurs et des informations relatives aux contrevenants.

IX. Bilan personnel

Créer une application Flutter se synchronisant avec un serveur Odoo m'a beaucoup enrichi, aussi bien au niveau de la programmation technique qu'au niveau de la conception théorique. Outre le fait que cela a été l'occasion pour moi de développer mes compétences en développement mobile, ce travail de fin d'étude m'a permis de m'initier au développement de modules Odoo malgré le fait que cette partie du projet n'ait pas été aboutie (à cause de la situation du COVID-19).

Travailler sur ce projet, depuis l'analyse de l'existant, jusqu'aux phases de tests, m'a permis de mettre en pratique de nombreux concepts théoriques que j'ai appris à l'ECAM et de vivre toutes les étapes de démarrage d'un nouveau projet. J'ai eu notamment l'occasion d'interviewer des agents traitants des SAC, afin d'avoir une meilleure compréhension des cas d'utilisations, ce qui m'a permis d'avoir des commentaires et critiques extérieurs au début du projet.

Malheureusement, le fait que le projet ait été réalisé entièrement à distance durant toute la durée du TFE, a eu un grand impact sur l'avancé global du projet. En effet, je n'ai pu bénéficier d'une aide extérieure assez limitée comparée à la complexité du fonctionnement de l'ARP. J'ai également l'impression que ce travail aurait pu me permettre d'établir des liens avec d'autres développeurs si celui-ci avait pu se dérouler en présentiel ce qui m'aurait aidé à m'intégrer plus facilement dans le monde professionnel, chose également déploré par mon promoteur.

X. Bibliographie

[Noviat SA, s.d.] Noviat SA : « Noviat », en ligne : <https://www.noviat.com/>

[Kevin Riga, 24 novembre 2016] Kevin Riga : « Les sanctions administratives communales (SAC) : de l'amende aux sanctions alternatives », en ligne : <https://www.police.be/5328/actualites/les-sanctions-administratives-communales-sac-de-lamende-aux-sanctions-alternatives>

[Commune de Saint-Josse, s.d.] Commune de Saint-Josse : « Sanctions administratives communales (SAC) », en ligne : <http://sjtn.brussels/fr/la-commune/prevention-securite/sanctions-administratives-communales-sac>

[UVCW, s.d.] UVCW : « Les sanctions administratives communales », en ligne : <https://www.uvcw.be/police-administrative/focus/art-2297>

[Moniteur belge, 25 septembre 2018] Moniteur belge : « Loi relative aux sanctions administratives communales », en ligne : https://www.ejustice.just.fgov.be/cgi_loi/change_lg.pl?language=fr&la=F&cn=2013062404&table_name=loi

[Ihor Feoktistov, s.d.] Ihor Feoktistov : « Top 8 Flutter Advantages and Why You Should Try Flutter on Your Next Project », en ligne : <https://relevant.software/blog/top-8-flutter-advantages-and-why-you-should-try-flutter-on-your-next-project/>

[Kate Ta, 6 octobre 2020] Kate Ta : « Flutter Advantages: 10 Reasons Why Using Flutter For Your Next Project », en ligne : <https://agiletech.vn/flutter-advantages-disadvantages-for-business/>

[Odoo SA, s.d.] Odoo SA : « Building a module » in *Developer Doc*, en ligne : <https://www.odoo.com/documentation/9.0/howtos/backend.html>

XI. Annexes

Dépôt application Flutter : https://github.com/16050/sac_mobile

Dépôt module Odoo : https://github.com/16050/SAC/tree/master/sac_module