République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed El Bachir El Ibrahimi - Bordj Bou Arréridj-
Faculté des Mathématiques et d'informatique

Département d'informatique

MEMOIRE

Présenté en vue de l'obtention du diplôme

Master en informatique

Spécialité : **Réseaux et Multimédia & Informatique Décisionnelle**

THEME

Simulateur de bracelet électronique de surveillance des condamnés sous Android et iOS

σκιά
Skià pour le placement sous surveillance mobile

Présenté par :

MERNIZ Hichem

MERNIZ Abdelkader

Devant le jury composé de :

Président

Examinateur

Encadreur Mme: SAIDANI Kaouther MCB à L'U. El Bachir El Ibrahimi- BBA

Promotion: 2020/2021

Remerciements

*Nous tenons tout d'abord à remercier **ALLAH** le tout puissant, qui nous a donné la force et la patience d'accomplir ce modeste travail.*

*En second lieu, nous tenons à remercier notre promotrice Mme **SAIDANI Kaouther**, pour son orientation, sa confiance, sa patience, qui ont constitué un apport considérable sans laquelle ce travail n'aurait pas pu être mené au bon port. Qu'elle trouve dans ce travail un hommage vivant à sa haute personnalité.*

*Nos vifs remerciements vont également aux **Membres du Jury** pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.*

Nous tenons à exprimer nos sincères remerciements à tous les ***Professeurs*** *qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.*

*Enfin, nous tenons également à remercier nos **Familles,** nos **amies** et toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.*

Table des matières

Chapitre 3 Architecture et Modélisation

Chapitre 4 Implémentation

Liste des Figures

Liste des Tableaux

Liste des Abréviations

SE : Surveillance électronique

PSE : Placement sous surveillance électronique

PSEM : Placement sous surveillance électronique mobile

REST: Representational State Transfer

API: Application Programming Interface

http: Hypertext Transfer Protocol

Https : Hypertext Transfer Protocol Secure

UML: Unified Modeling Language

MVC: Model-view-controller

JWT: Json Web Token

JWS: JSON Web Signature

JWE: JSON Web Encryptions

Chapitre 1

Introduction générale et problématique

1.1 Introduction

Ce chapitre présente le contexte général dans lequel s'inscrit notre travail «Simulateur de bracelet électronique de surveillance des condamnés sous Android et iOS». Dans un premier temps nous allons présenter la définition et l'utilisation de la biométrie. Par la suite, nous aborderons les particularités de la surveillance électronique comme outil de gestion des peines carcérales –bracelet électronique de surveillance des condamnés - auxquelles nous nous intéressons, puis nous définirons la problématique et les objectifs de notre projet. Enfin nous décrirons notre contribution.

1.2 Contexte

Notre travail rentre dans le cadre d'un projet de recherche visant a développer un simulateur de bracelet électronique de surveillance des condamnés sous Android qui a pour rôle la collecte et la transmission régulière des empreintes digitales des prisonniers à un centre de contrôle, dans le but de s'assurer de la présence de ces derniers à chaque instant et en tous lieux. Le travail porte sur l'amélioration des applications mobiles de surveillance des prisonniers sous Android qui consistent à contrôler via un smartphone si effectivement un condamné est chez lui ou pas. L'application doit être hébergée dans un serveur distant. Un smartphone est délivré au condamné avec l'application client installée. Via la connexion internet, on demande au condamné de se connecter au serveur pour enregistrer ses empreintes digitales. De façon régulière, le condamné doit activer sa présence en posant juste le doigt sur son smartphone pour la lecture et la vérification d'empreintes.

Pour ce projet, deux domaines sont à prendre en compte : la biométrie et la surveillance électronique (SE).

1.2.1 La biométrie

La biométrie désigne l'ensemble des procédés de reconnaissance, d'authentification et d'identification d'une personne par certaines de ses caractéristiques biologiques (comme l'ADN), comportementales (comme la dynamique de la signature) ou morphologiques (comme l'empreinte digitale), uniques et propres à chaque individu [1].

Les caractéristiques biométriques par lesquelles il est possible de vérifier l'identité d'un individu sont appelées modalités biométriques. La FIGURE 1-1 illustre un exemple de quelques modalités biométriques. Ces modalités sont basées sur l'analyse des données liées à l'individu et sont généralement classées en trois catégories : biométrie biologique, comportementale et morphologique [1].

Cette technologie de pointe est devenue en quelques années le moyen le plus fiable d'identification d'une personne car elle comporte un avantage primordial sur les solutions d'authentification traditionnelles compte tenu de *la relation forte entre l'authentifiant et l'utilisateur*.

Les applications biométriques sont nombreuses et permettent d'apporter un niveau de sécurité supérieur en ce qui concerne les accès logiques (ordinateurs, comptes bancaires, données sensibles, etc.) ou des accès physiques (bâtiments sécurisés, aéroports, etc.) [1].

La biométrie s'invite progressivement dans notre vie quotidienne, elle fait partie des grands enjeux pour un monde plus sûr. Le marché des produits d'authentification et d'identification est en pleine croissance, dû à la nécessité croissante du besoin de sécurité de chacun, dans les domaines privés, professionnel ou public [1].

Une technologie établie, avec plusieurs technologies : empreintes digitales, visage 2D, Visage 3D, iris, rétine, voix, réseau veineux, forme de la main, comportemental (signature dynamique, frappe au clavier, navigation sur une tablette ou smartphone, façon de marcher). Pour un système d'authentification encore plus robuste, on peut associer simultanément plusieurs méthodes biométriques (multimodales).

Les exigences de sécurité de la société d'aujourd'hui ont placé la biométrie au centre d'un large débat car elle est en train de devenir un élément clé dans une multitude d'applications. Elle vient remplacer ou renforcer les dispositifs à clé, à mot de passe ou à badges pouvant présenter des failles en matière de sécurité [1].

A l'heure actuelle, l'empreinte digitale est la solution biométrique la plus répandue et la plus connue du grand public mais ce n'est pas la seule [1].

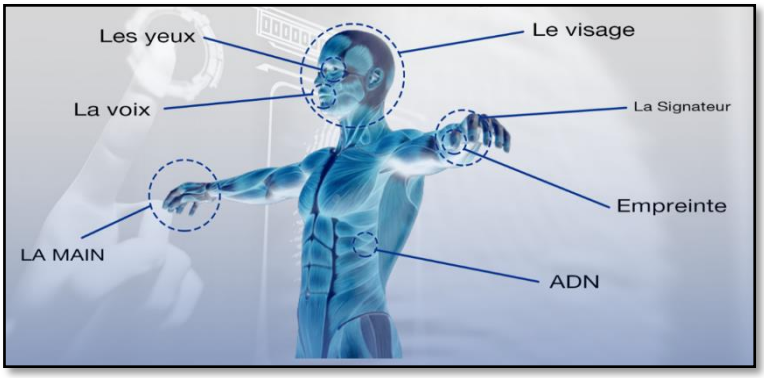


FIGURE 1-1- Les modalités biométriques (physiques et comportementales)

Pour ce projet, une modalité biométrique est à prendre en compte : l'empreinte digitale.

1.2.1.1 La modalité empreinte digitale

D'après les archives historiques, cette modalité semble être la plus ancienne car des traces qui datent de plus de 4000 ans ont été découvertes en Egypte. Cette modalité fut aussi utilisée très tôt par les Chinois pour signer des documents officiels. La puissance de cette modalité réside dans le fait que le dessin formé par les empreintes est unique pour chaque personne [2].

En 1901 la technique d'authentification au moyen des empreintes fut adoptée officiellement en Angleterre dans le système judiciaire.

Cette technique fut ensuite largement développée dans les enquêtes criminelles et permit de résoudre un bon nombre d'affaires. De nos jours les empreintes sont toujours largement utilisées et reconnues comme méthode d'identification fiable [2].

Les systèmes de vérification procèdent en général à l'extraction des caractéristiques principales de l'empreinte telles que les bifurcations de crêtes, les terminaisons, le centre etc.... et les utilise pour l'authentification FIGURE 1-2 [2].

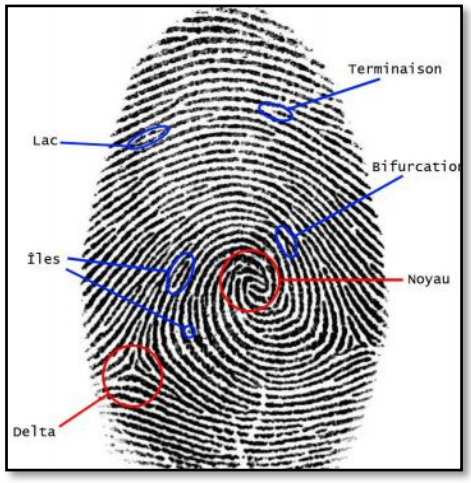


FIGURE 1-2- Les points caractéristiques de l'empreinte [2]

Une empreinte digitale est constituée d'un ensemble de lignes localement parallèles formant un motif unique pour chaque individu FIGURE 1-3, on distingue [2]:

- *Les crêtes (Ridges en anglais)* : ce sont les lignes en contact avec une surface au toucher.
- *Les vallées (valleys en anglais)* : ce sont les creux entre deux crêtes.

Chaque empreinte possède un ensemble de points singuliers globaux (les centres et les deltas) et locaux (les minuties) [2]:

1- **Les minuties** : représentent des discontinuités locales et marquent les positions où la crête se termine ou bifurque. Cela constitue les types de minutie les plus fréquentes, bien qu'un total de 16 types de minuties FIGURE 1-4 ait été identifié.

Chaque minutie peut être décrite par un nombre d'attributs tels que la position(x,y), l'orientation et le type (terminaison ou bifurcation).

2- Les points singuliers (centres et les deltas) :

- *Le centre (Core):* le centre est le lieu de courbure maximale des lignes d'empreinte les plus internes. Il est aussi appelle le point core.
- *Les deltas (Delta):* un delta est proche du lieu où se s'éparent deux lignes d'empreintes vérifiant les propriétés suivantes : ces lignes se séparent suivant deux directions orthogonales et sont les lignes les plus internes vérifiant la propriété précédente.

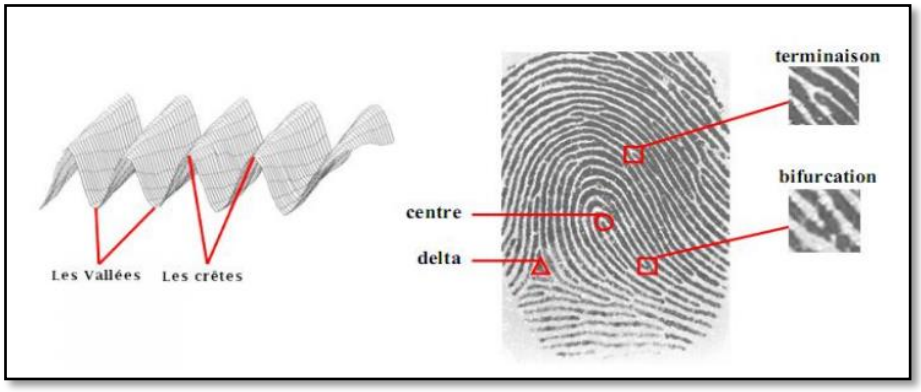
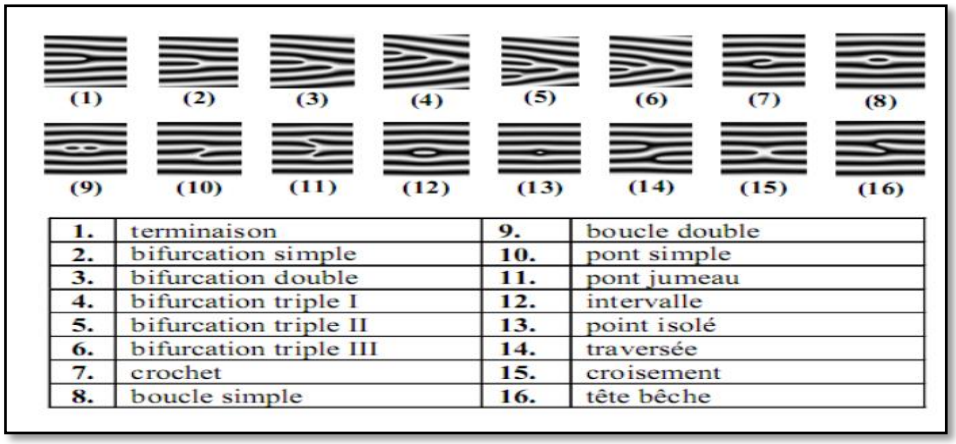


FIGURE 1-3 - Caractéristiques d'une empreinte digitale [2]



1.	terminaison	9.	boucle double
2.	bifurcation simple	10.	pont simple
3.	bifurcation double	11.	pont jumeau
4.	bifurcation triple I	12.	intervalle
5.	bifurcation triple II	13.	point isolé
6.	bifurcation triple III	14.	traversée
7.	crochet	15.	croisement
8.	boucle simple	16.	tête bêche

FIGURE 1-4 – Les 16 types des minuties [2]

La position et le nombre de centres et de deltas permettent de classifier les empreintes en catégorie selon leur motif général, on distingue principalement trois grandes familles FIGURE 1-5 [2].

- *Les boucles (loop)* représentent 65% des empreintes rencontrées.
- *Les spires (whorl)* représentent 30% des empreintes rencontrées.
- *Les arches (arch)* représentent 5% des empreintes rencontrées.

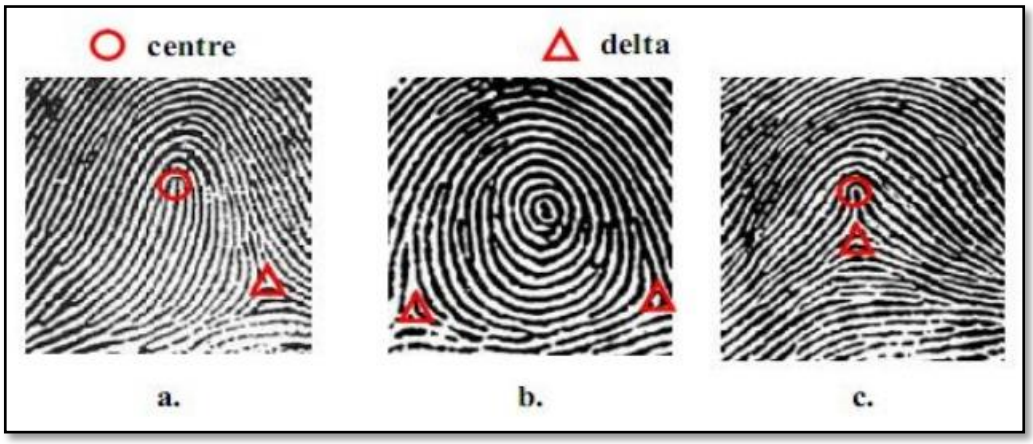


FIGURE 1-5 – Les trois principales classes d'empreinte, boucle (a), spire (b), arche (c) [2]

L'ensemble formé par la disposition des points singuliers constitue un motif unique pour chaque individu, en effet il a été montré [3] que l'empreinte digitale se forme au cours du troisième mois de la vie vitale, le motif général est influencé par les gènes héréditaires mais l'apparition des détails (minuties) est créée de manière accidentelle par des pressions variables aléatoires sur les surfaces tactiles. Ainsi l'empreinte est unique pour tout individu, *y compris pour des vrais jumeaux* [4] *et il a été montré que les méthodes de reconnaissance actuelles permettent d'identifier efficacement les jumeaux* [7]. De plus les empreintes une fois formées ne changent plus au cours de la vie d'une personne, ces deux caractéristiques en font un moyen de reconnaissance très efficace.

Les principaux avantages et inconvénients de cette modalité sont présentés au Tableau 1.1

Tableau 1.1- Avantages et inconvénients des empreintes digitales

Avantages	Inconvénients
Technique la plus mûre, utilisée, maîtrisée et reconnue.	Lecture compromise si les doigts sont sales ou abimés.
Taille et prix du capteur abordable.	Risques de contamination par contact.
Facile à utiliser avec un traitement rapide.	Possibilités de fraude par moulage du doigt.
Caractérisation unique de la personne.	
L'arrangement d'arêtes demeure permanent durant toute la vie.	

1.2.1.2 Fonctionnement et Architecture générale d'un système biométrique

Un système biométrique peut être représenté par quatre modules principaux [5]:

- *Module d'acquisition (capteur)* : destiné à l'acquisition des données biométriques d'une personne. Cela peut être un appareil photo, un lecteur d'empreintes digitales, une caméra de sécurité, etc…
- *Module d'extraction de descripteurs*: Ce module traite les données acquises et procède à l'extraction des valeurs caractéristiques nécessaires à l'opération d'authentification.
- *Module de comparaison* : Ce dernier procède à la comparaison des données acquises avec ceux enregistrées au préalable durant la phase de collecte d'informations appelées (enrôlement). Le résultat de cette comparaison est concrétisé par un score (S) à utiliser par le module de décision.

- *Module de décision* : Ce module est responsable de l'acceptation ou du rejet de l'identité d'une personne en comparant le score (S) généré par le module de comparaison avec un seuil de sécurité (M) donné. Dans le cas ou (S) > (M), l'individu sera accepté, dans le cas contraire il est rejeté.

Les systèmes biométriques fonctionnent selon trois modes que sont l'enrôlement, la vérification d'identité et l'identification [5]:

- *Le mode d'enrôlement* : L'enrôlement est la première phase de tout système biométrique. Il s'agit de l'étape pendant laquelle un utilisateur est enregistré dans le système pour la première fois. Pendant l'enrôlement, la caractéristique biométrique est mesurée en utilisant un capteur biométrique afin d'extraire une représentation numérique. Cette représentation est ensuite réduite, en utilisant un algorithme d'extraction bien défini, afin de réduire la quantité de données à stocker pour faciliter la vérification et l'identification.

- *Le mode de vérification ou authentification* : est une comparaison "1 :1", dans lequel le système valide l'identité d'une personne en comparant les données biométriques saisie avec le modèle biométrique de cette personne stockée dans la base de données du système. Dans ce cas, le système renvoie uniquement une décision binaire (oui ou non).

- *Le mode d'identification* : est une comparaison "1:N", dans lequel le système reconnaît un individu en l'appariant avec un des modèles de la base de données. La personne peut ne pas être dans la base de données. Ce mode consiste à associer une identité à une personne.

1.2.1.3 Structure d'un système complet de reconnaissance d'empreinte

Un système automatique complet de reconnaissance d'empreintes digitales est une chaîne de processus qui à partir du doigt d'un utilisateur en entrée renvoie un résultat en sortie, permettant ainsi à l'utilisateur d'accéder ou non à des éléments nécessitant une protection [6].

La réalisation d'un tel système a fait l'objet de très nombreuses recherches et des méthodes très différentes de traitement ont été proposées [7]. Néanmoins ces systèmes répondent toujours à la même structure (FIGURE 1-6).

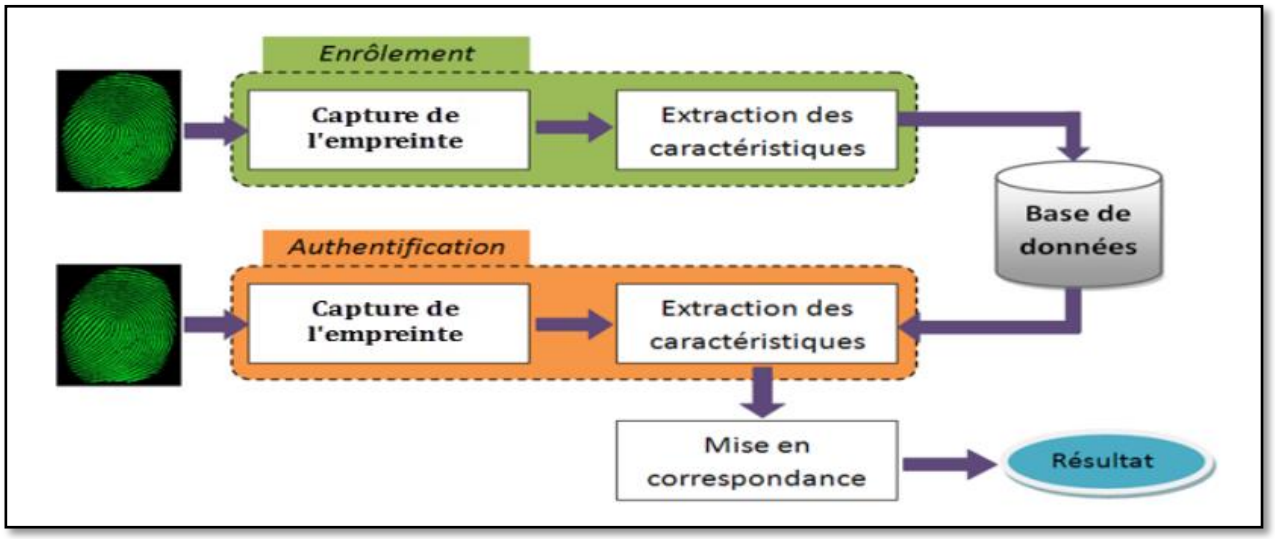


FIGURE 1-6 – Architecture générale d'un système de reconnaissance d'empreinte digitale

La première phase permet d'obtenir une image de l'empreinte de l'utilisateur (acquisition), laquelle va subir un prétraitement pour extraire l'information utile de l'image (gabarit, ou Template en anglais) suivi éventuellement d'un traitement supplémentaire permettant d'éliminer de possibles fausses informations qui se seraient glissées entre temps dans la chaîne de traitement. Ensuite si l'utilisation du système consiste juste à créer une base de données (stockage, Enrôlement) le Template est éventuellement compressée puis stockée dans la base de données au moyen d'une technique d'archivage.

Pour un système d'identification l'ensemble des empreintes présentes dans la base de données pouvant correspondre à celle de l'utilisateur (modèle identique) sont désarchivées et comparées (appariement, Matching) une à une avec celle de l'utilisateur, si une éventuelle correspondance est trouvée des informations personnelles concernant l'utilisateur sont renvoyées par le système.

- ***L'acquisition de l'empreinte***

La première phase d'un système de vérification consiste à obtenir une image de l'empreinte du doigt. Longtemps le seul moyen existant a été l'utilisation du papier et de l'encre ce qui a rendu la tâche de reconnaissance très lourde. En effet la qualité de l'image était plutôt mauvaise (plusieurs acquisitions étaient nécessaires) et l'extraction du Template était effectuée visuellement par un expert (processus très long et pénible). Heureusement avec le développement de l'informatique et de la microélectronique de nouveaux moyens d'acquisition ont fait leur apparition, permettant ainsi d'accélérer la chaîne de traitement en l'automatisant. Aujourd'hui, un bon nombre de capteurs d'empreintes existent sur le marché [6]. Ils se distinguent, notamment par :

- ✓ Leur technologie.
- ✓ Leur coût.
- ✓ Leur qualité d'acquisition.
- ✓ Leur facilité d'intégration (téléphone, ordinateur portable, etc.)
- ✓ leur capacité à détourner les moulages d'empreintes.

Pour permettre une reconnaissance fiable, un prétraitement est alors nécessaire pour extraire l'information utile de l'image suivi éventuellement d'un post-traitement supplémentaire permettant d'éliminer de possibles les fausses informations qui se seraient glissées entre temps dans la chaîne de traitement.

- ***Le stockage et la phase d'appariement***

Pour les systèmes disposant de grosses bases de données, l'identification peut poser un problème en temps de calcul si le Template d'entrée doit être comparée avec toutes les Templates présentes dans la base. C'est pourquoi un processus de classification et de déclassification est nécessaire pour limiter les temps de recherche [6].

Lorsqu'une image est stockée, un groupe spécifique lui est attribué en fonction de ses caractéristiques. Lors de l'identification on désarchive l'ensemble des Templates de la base correspondant au groupe de l'empreinte nécessitant l'identification.

Puis chacune des images désarchivées est comparée avec celle de l'utilisateur. Ceci permet de réduire sensiblement les temps de recherche en limitant le nombre d'images à comparer, à condition que les différentes catégories soient judicieusement choisies. Parmi les différentes techniques existantes on distingue principalement l'extraction des singularités de l'image (la position des centre et delta permet de déterminer la classe de l'empreinte) et l'utilisation des réseaux de neurones...etc [6].

La phase d'appariement est l'étape critique du système, elle reçoit en entrée deux Templates issues de deux acquisitions différentes d'empreinte et renvoie en sortie un résultat binaire indiquant si oui ou non les deux Templates proviennent de la même empreinte. Bien entendu deux empreintes provenant de la même personne ne seront jamais exactement identiques en raison de l'élasticité de la peau, de la présence de poussière, de l'orientation du doigt lors de l'acquisition. La phase d'appariement va donc calculer le degré de similarité (taux d'appariement) entre les deux Templates et décider si elles peuvent être considérées identiques en fonction d'une valeur de seuil [6].

Bien que les deux empreintes puissent être comparées directement par corrélation, la méthode qui a suscité le plus d'intérêt utilise les caractéristiques locales des minuties et consiste en l'appariement basé sur l'alignement d'un motif de point car il est simple en théorie, efficace pour faire face à la fausse information détectée dans les phases précédentes, et rapide par rapport aux autres méthodes. Cet algorithme est divisé en deux processus [6]:

- ✓ *L'alignement*: on évalue la transformation géométrique (orientation, translation, homothétie) entre les deux ensembles à traiter et on les aligne suivant cette transformation.
- ✓ *L'appariement*: on évalue le nombre d'éléments caractéristiques qui sont alignés (moyennant une certaine marge d'erreurs car un alignement parfait est impossible) et le taux d'appariement est calculé en fonction des correspondances rencontrées.

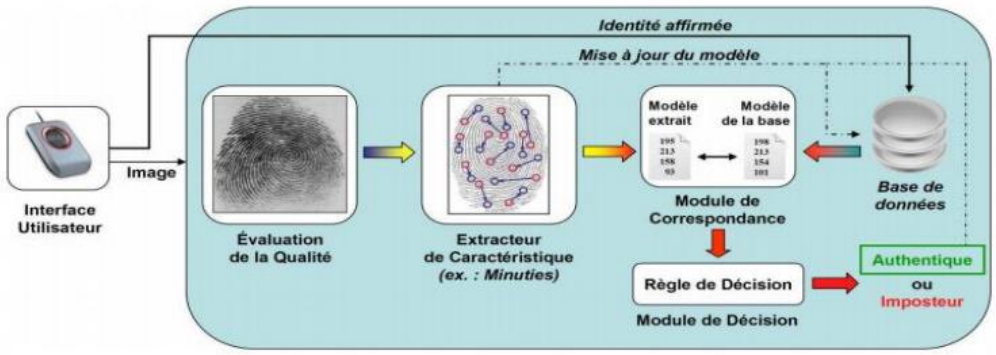


FIGURE 1-7 – Authentification d'un individu dans un système biométrique [6]

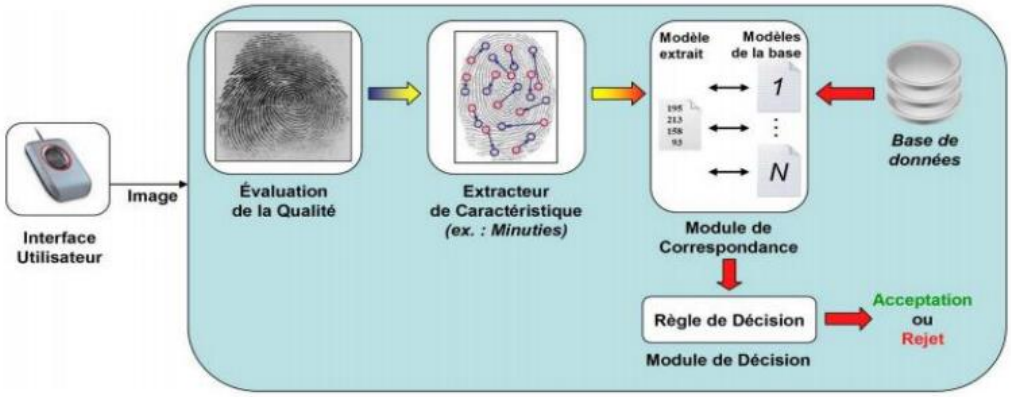


FIGURE 1-8 – Identification d'un individu dans un système biométrique [6]

1.2.2 La surveillance électronique (SE)

La surveillance des prisonniers à domicile par voie électronique est un phénomène récent. La surveillance électronique comme outil de gestion des peines carcérales, introduite en France au milieu des années 1990, consiste à astreindre à leur domicile les condamnés équipés d'un système de surveillance électronique. Les horaires d'entrée et de sortie du domicile sont fixés par le juge d'application des peines (JAP) à qui les détenus doivent rendre des comptes via les surveillants du pôle, de surveillance électronique d'une part, et leur conseiller pénitentiaire d'insertion et de probation (CPIP) d'autre part [8].

La surveillance électronique est utilisée comme solution alternative à l'emprisonnement, ou à la libération conditionnelle, un système de surveillance qui comprend une ligne téléphonique résidentielle, un dispositif de surveillance et un bracelet à la cheville, qui suit le mouvement du condamné et vérifier où se trouve la personne et détermine si elle viole les conditions fixées par les tribunaux. Cela signifie qu'il doit rester dans un endroit fixe (sa maison) pendant certaines heures de la journée. La surveillance par bracelet électronique vise à réduire l'utilisation de la prison, réduire le budget de l'état pour les prisons et à soutenir l'élimination des crimes [8].

1.3 Problématique

Le placement sous surveillance électronique (PSE) ou 'bracelet électronique' est une mesure d'aménagement de peine permettant d'exécuter une peine d'emprisonnement sans être incarcéré, généralement fixé à la cheville, il est posé au greffe de l'établissement pénitentiaire ou au SPIP (service pénitentiaire d'insertion et de probation). Un surveillant installe dans le logement un boîtier qui se branche sur la prise de courant. Le condamné placé sous surveillance électronique est assigné à résidence, chez lui ou chez un hébergeur, pendant les heures fixées par le juge [8].

Le bracelet et son boîtier, relié à la ligne téléphonique, ne doit jamais être enlevé : le condamné se douche et dort avec. Le boîtier reçoit les informations émises par le bracelet.

Les plus gros problèmes avec le placement sous surveillance électronique (PSE) ou bien le bracelet électronique sont:

1. Le coût pour le condamné et sa famille. Le programme peut lourdement accabler les parents indigents à paient les frais d'installation, des frais journalier pour l'utilisation de l'équipement, et le coût de l'équipement endommagé. Si une famille ne peut pas se permettre de payer les frais associés à la surveillance électronique, le service peut être résilié, ce qui discrimine la famille indigente.

2. Les détenus sont soumis à une grave violation de leur vie privée et à l'inviolabilité de leur résidence, car ils peuvent recevoir la visite de l'agent de probation à tout moment.

3. Mettre l'appareil électronique sur le corps a des mauvais effets sur la santé, en plus des effets psychologiques résultant de la société, où il ne peut pas effectuer des activités où le bracelet est visible (ne doit jamais être enlevé : le condamné se douche et dort avec).

4. Le bracelet électronique est statique (il ne fait que constater si la personne est ou non à son domicile, car le dispositif ne permet pas de localiser la personne lorsqu'elle est libre de sortir).

A cause de ces problèmes, on a proposé l'idée de créer un simulateur de bracelet électronique de surveillance des condamnés sous Android et iOS.

1.4 Objectifs

Dans le cadre de ce travail, notre objectif consiste à réaliser un simulateur de bracelet électronique de surveillance des condamnés sous Android et iOS pour :

- Réduire les dépenses engagées par les détenus lors de l'utilisation du bracelet électronique par l'utilisation d'une application mobile sur un smartphone.
- Protéger la vie privée des détenus par l'utilisation d'une application mobile sous Android et iOS au lieu d'utiliser un bracelet visible.
- Garder l'humanité des condamnés et éviter les mauvais effets sur la santé résultant de l'utilisation du bracelet électronique fixé à la cheville et les effets psychologiques résultant de la vision de la société (le bracelet électronique fixé à la cheville ne doit jamais être enlevé : le condamné se douche et dort avec).

- Offrir le plus haut degré de suivi (l'application permet de savoir où se trouve le condamné à chaque instant et en tous lieux car la personne reste contrôlée dans tous ses déplacements lorsqu'elle est sortie) d'une part et un haut degré de sécurité par l'utilisation de la modalité biométrique 'empreinte digitale' (unique pour chaque individu) d'autre part.

1.5 Contribution

Dans le cadre de ce projet, on a essayé d'analyser l'effet de l'utilisation du bracelet électroniques comme technologie de surveillance des prisonniers, cette dernière présente comme on l'a déjà montré beaucoup inconvénients, ceci nous a poussé à essayer de trouver une solution alternative qui peut résoudre notre problématique.

Afin d'améliorer le système de surveillance des détenus, on veut qu'on introduit la *technologie biométrique*, cette dernière peut être la clé pour supprimer les méfaits du bracelet électronique.

En effet l'utilisation d'une *application mobile sous Android et iOS* va augmenter considérablement le niveau de suivi et le niveau de sécurité vu qu'elle utilise des paramètres biologiques uniques qui distinguent chaque individu.

En plus de ce qu'on a mentionné, *l'application mobile sous Android et iOS* est une solution envisageable pour ne pas discriminer les familles des détenus incapables de payer les frais d'installation d'un bracelet électronique.

Finalement notre projet est en mesure de respecter la vie des gens, de supprimer l'effet psychologique et d'éviter les mauvais effets sur la santé résultant de l'utilisation des bracelets électroniques ((le bracelet électronique fixé à la cheville ne doit jamais être enlevé : le condamné se douche et dort avec).

1.6 Plan du mémoire

Après une description globale du contexte, de la problématique, des objectifs et de la contribution de notre travail nous nous focaliserons **en deuxième partie** sur la description du bracelet électronique comme un modèle existant de la surveillance électronique des condamnés et nous présenterons une description sommaire de notre modèle proposé.

En troisième partie nous présenterons la modélisation et la conception de notre modèle proposé qui consiste à simuler sous Android et iOS le bracelet électronique pour surveiller les prisonniers à chaque instant et en tous lieux par l'utilisation d'une application mobile d'une part et pour augmenter le niveau de sécurité par l'utilisation de la modalité empreinte digitale d'une autre part. **La quatrième partie** sera consacrée à la description des outils et langage utilisés et à la présentation des résultats obtenus. Enfin nous terminerons par une conclusion et des perspectives.

Chapitre 2

Etat de l’art

2.1 Introduction

Dans le cadre de développement d'un système de surveillance pour les condamnés, il est important d'examiner les modèles existants et de noter par la suite, leurs limites en pratiques, le modèle existant le plus connus est le bracelet électronique. Dans un premier temps nous allons présenter le modèle bracelet électronique de surveillance des condamnés dans l'objectif de comprendre leur fonctionnement et de montrer ces inconvénients et ces limites, ensuite nous donnerons une description sommaire de notre modèle proposé.

2.2 Modèles existants

2.2.1 Le placement sous surveillance électronique (PSE) ou 'bracelet électronique'

Le bracelet électronique est une mesure d'aménagement de peine d'une personne condamnée à de la prison. La personne s'engage à rester à son domicile à certaines heures, l'administration pénitentiaire contrôlant le respect de ses obligations à l'aide du bracelet qu'elle porte sur elle. Le placement sous surveillance électronique peut être prononcé en vue d'éviter l'incarcération d'une personne condamnée. Cette mesure peut également être prise dans le cadre de la remise en liberté d'une personne condamnée pour certains faits à une longue peine d'emprisonnement [9].

Le bracelet électronique vise à favoriser la réinsertion par un accompagnement et un contrôle du respect des obligations fixées par le juge de l'application des peines. Il peut par exemple permettre à la personne d'exercer une activité professionnelle ou de suivre une formation. Toutes les personnes condamnées ne peuvent pas bénéficier du bracelet électronique. Il faut que la peine (ou le cumul des peines) ou que la durée de la peine restant à effectuer soit inférieure ou égale à 2 ans d'emprisonnement, délai réduit à 1 an en cas de récidive [9].

Le bracelet électronique peut également être accordé à un détenu en cas de liberté conditionnelle. Il peut également être accordé à un mis en examen dans le cadre d'une assignation à résidence, alternative à la détention provisoire [9].

Le bracelet électronique est porté à la cheville. Il comprend un boîtier GPS. Ce dispositif, géré par l'administration pénitentiaire, permet de s'assurer en temps réel, via la localisation, que la personne n'enfreint pas les interdictions fixées par le juge. Si tel est le cas, une procédure d'alerte se met immédiatement en place. Les horaires sont fixés par le juge. Ils correspondent généralement aux heures où la personne placée sous surveillance ne travaille pas (hors temps de trajet). Exemple : la personne peut être contrainte de rester à son domicile entre 20h et 7h du matin [10].

Ce dispositif se fixe à la cheville. Il contient une puce électronique qui permet de le localiser grâce à un système de géolocalisation. L'autorité de surveillance reçoit une alarme dans le cas où la personne surveillée ne respecte pas ses obligations, comme rester à son domicile à certaines heures par exemple [10].

L'installation du dispositif au domicile est le moment où se constitue la mise sous écrou du condamné. Un représentant de la puissance publique, l'agent pénitentiaire, pénètre le lieu de vie privée du condamné et fait la rencontre de ses proches. Les modalités concrètes de la peine sont stipulées : usage du matériel de surveillance et relations possibles avec le pôle de surveillance électronique régional, délimitation de la portée du dispositif par rapport au lieu de vie. L'agent du Pôle de Surveillance Électronique détermine avec le condamné le périmètre capté par le boîtier récepteur, et le surveillé fait le tour de l'appartement, vaque d'une pièce à l'autre, sort sur le palier, sur son balcon ou dans le jardin pendant que le surveillant configure la portée d'onde [10].

Les bracelets de la nouvelle génération peuvent également recueillir des données d'ordre physiologique, le rythme cardiaque du porteur ou son taux d'alcool dans le sang par exemple, comme cela se fait déjà en Suisse. Des bracelets fabriqués par Geosatis y ont été mis en place pour les personnes condamnées pour conduite en état d'ivresse ayant l'obligation de se sevrer [11].

Le bracelet calcule le taux d'alcoolémie de façon aléatoire ou alors à un horaire préalablement fixé, et une alarme se déclenche si les taux mesurés dépassent le seuil fixé [11].

2.2.1.1 Fonctionnement

Après une enquête de faisabilité technique, un rapport circonstancié du SPIP et l'accord du JAP, les agents PSE accompagnent la personne (dans le cadre d'une sortie de détention) ou la retrouvent chez elle pour installer le matériel, expliquer son fonctionnement et rappeler également les obligations qui en découlent [12].

1- **Unité fixe** Ce boîtier noir, avec un combiné de téléphone et un écran tactile, est "relié" au bracelet électronique qui lui transmet les informations sur sa position. C'est aussi un moyen pour le pôle centralisateur de surveillance de la région, chargé de la surveillance 24 h sur 24 et 7 jours sur 7 des personnes sous PSE, de communiquer avec la personne via le combiné, notamment si l'alarme se déclenche. Et vice versa.

Via un écran tactile la personne surveillée a accès à un espace "calendrier" lui rappelant ses horaires de sortie autorisé fixés par le Jap ou encore les numéros importants à connaître (SPIP, pôle de surveillance...). Notamment, en cas de problème technique (panne, coupure de courant...) ou en cas d'absence pour motif grave hors des horaires autorisés.

2- **Le bracelet** Avant toute installation, le surveillant prend la mesure de la cheville de la personne afin que le bracelet soit adéquat. Ce dernier est ensuite installé et activé. Étanche, il peut être porté sous la douche et ne sonne pas aux portiques des magasins. Il est, bien sûr, formellement interdit de l'enlever, ou même d'essayer, ce qui serait considéré comme une évasion. Il devra également être rendu en bon état. Toute dégradation du matériel (unité ou boîtier) est facturée à la personne concernée par le dispositif.

3- **La zone d'assignation** Quand le matériel est branché et installé, il est nécessaire de paramétrer la zone d'assignation dans laquelle la personne sous surveillance devra évoluer au quotidien, en dehors de ses horaires de sortie autorisés. Pour cela, la personne sous surveillance doit longer, lentement, les murs de chaque pièce du logement. Il est possible que la zone ne comprenne pas la boîte aux lettres et le local poubelle : la personne ne peut donc s'y rendre que dans le cadre de ses horaires de sortie.

4- **En cas d'alarme** L'alarme peut se déclencher si la personne ne respecte pas ses horaires de sortie (sortie trop tôt ou rentrée trop tardive) ou encore si l'unité fixe est débranchée ou déplacée. Dans ces cas-là le pôle centralisateur de surveillance appelle sur l'unité fixe ou sur son portable la personne pour savoir ce qu'il se passe. L'alarme est communiquée aux agents PSE qui contactent à leur tour la personne sous surveillance. Alarme qui sera également transmise au juge d'application des peines (Jap) qui peut alors prendre des sanctions.

5- **Cas spéciaux** Si la personne doit déménager alors qu'elle fait toujours l'objet d'un PSE, elle doit en informer, largement en amont, son conseiller pénitentiaire d'insertion et de probation et avoir l'accord du Jap. Un surveillant pénitentiaire, agent PSE, installera le dispositif dans le nouveau logement. Même parcours si elle doit faire changer ses horaires de sortie autorisées, notamment dans le cadre d'une recherche d'emploi et d'un nouvel emploi. Une demande doit être faite auprès du Spip qui la transmet au magistrat.

2.2.2 Le placement sous surveillance électronique mobile (PSEM)

Le placement sous surveillance électronique mobile (PSEM) est une technologie permettant de savoir où se trouve la personne placée à chaque instant et en tous lieux.

Il se distingue ainsi du PSE statique qui permet uniquement de savoir si une personne est bien en un lieu donné – son domicile – pendant des périodes déterminées (afin de permettre au condamné de quitter son domicile pour travailler, suivre une formation, etc.). La géolocalisation (GPS) du PSEM permet même la transmission d'une alarme en temps réel au centre de contrôle en cas de violation des interdictions prononcées (approcher une école primaire, par exemple) [10].

2.2.2.1 Les différents modes de surveillance du (PSEM)

1- *Le mode semi-actif* : il permet non seulement de fournir un rapport quotidien des déplacements des placés, mais aussi d'émettre une alarme dès que le placé ne respecte pas ses obligations. C'est uniquement lorsque l'alarme de violation des obligations ou interdictions fixées est émise que ce mode permet de suivre en quasi temps réel les déplacements de la personne.

Le journal ou rapport quotidien des déplacements de chaque placé peut prendre la forme d'une cartographie des déplacements de l'intéressé avec indication des horaires et des vitesses de déplacement si l'administration pénitentiaire le demande [10].

L'alarme se déclenche dans les cas suivants [10]:

- violation des lieux interdits. L'alarme est émise avant que le placé ne pénètre dans ce lieu interdit (grâce à la programmation des zones « tampons ») et se poursuit tant qu'il n'est pas sorti du périmètre interdit. Le personnel pénitentiaire étant alerté dès que le placé entre dans une zone tampon, il peut avertir ce dernier du risque qu'il prend et lui donner les instructions qui s'imposent;
- non respect des horaires d'assignation ;
- tentative du placé de retirer le bracelet (grâce à la détection thermique) ou de le détériorer.

Toute alarme de violation déclenche la rédaction d'un rapport précisant la nature de la violation, l'heure, le lieu, la cartographie des déplacements 4 heures avant l'alarme.

2- *Le mode passif* : il se limite à l'envoi quotidien aux autorités pénitentiaires d'un journal des déplacements du placé.

3- *Le mode actif* : qui permet de suivre à tout moment, en direct, les déplacements du placé même quand celui-ci respecte ses obligations, n'a pas été retenu [10].

2.2.2.2 Matériels et fonctionnement

Le **bracelet** utilisé est du même type que le bracelet statique. Il est porté généralement à la cheville et est doté d'une batterie non rechargeable d'une durée de vie de 36m ois. Un système intégré avec détection thermique permet d'identifier les manipulations (ouverture, proximité du corps). Il émet en permanence un signale radio qui est capté par le boitier récepteur portable [10].

Le **récepteur portable** ou "**support GPS"** se porte à la ceinture lors des déplacements de la personne. Il dispose d'un GPS intégré : il reçoit en permanence les informations lui permettant de connaitre son positionnement. Il peut être rechargé : l'opération prend 3 à 5 heures suivant le matériel et donne au récepteur portable une autonomie d'environ 16 heures. Il dispose d'une fonctionnalité permettant au centre de surveillance de communique des messages que le porteur peut lire sur l'écran du récepteur [10].

Le **récepteur statique**, placé au domicile de la personne, complète la surveillance mobile et prend éventuellement le relais du récepteur portable qui peut alors être mis en veille, ce qui permet d'en économiser la batterie, soit branche afin d'être rechargé. D'installation très simple, le récepteur statique communique les messages au support GPS [10].

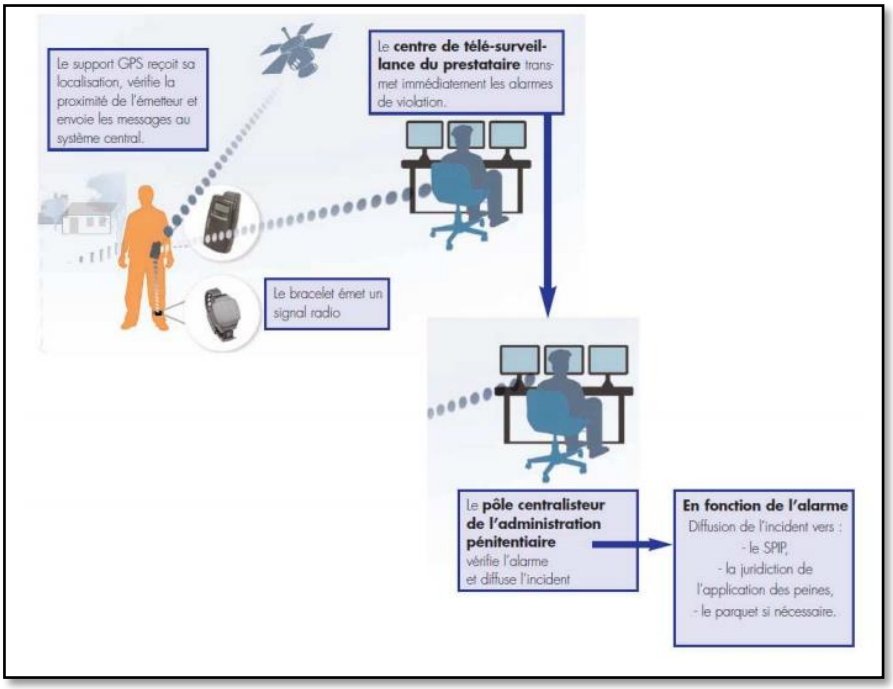


FIGURE 2-1- Fonctionnement du PSEM lors du déplacement [10]

2.3 Description sommaire du modèle proposé

Après avoir étudié les modèles existants, on a essayé de construire une description sommaire de notre application – Simulateur de bracelet électronique de surveillance des condamnés sous Android et iOS –

Notre modèle est basé sur la création d'un simulateur de bracelet électronique de surveillance sous Android et iOS nommé "SKIA pour le placement sous surveillance mobile", dédié aux prisonniers, Agents de police et Agents master, ce simulateur comprend :

1. *Une application mobile* pour le prisonnier qui devra être installé sur son téléphone portable, après la création d'une session (l'authentification), le condamné peut l'utiliser à tout moment en tous lieux de sa région. Elle conçue pour le pointage périodique des prisonniers par l'empreinte digitale, l'envoie de l'emplacement du condamné (en temps réel) au serveur du traitement et pour le reçu des notifications de la part de l'application elle mémé (locale) (exp : *Pointing !!! signal your presence*, *scan your fingerprint to authenticate*) et du serveur du traitement (exp : *Emplacement !!! you are in orange région*).

Remarque : Pour la communication entre *le serveur* et *le client* (application mobile) on a utilisé l'interface de programmation d'application (**REST API**) sous le protocole **HTTP/HTTPS** avec le **websocket**.

2. *Une application mobile* pour l'agent de police qui devra être installé sur son téléphone portable, après la création d'une session 'l'authentification', l'agent de police peut l'utiliser à tout moment dans sa région. Elle conçue pour suivre les prisonniers dans une région bien définie et recevoir des notifications (exp : le prisonnier dépasse la région verte, l'opération de pointage est échouée) de la part du serveur du traitement.

3. *Une application web* dédiée pour l'agent master. Elle conçue pour gérer (ajouter, modifier et supprimer) les prisonniers, les Agents de police et les régions, suivre les prisonniers (vérifier sa présence par l'utilisation d'un système de reconnaissance d'empreinte digitale) et pour recevoir des notifications de la part du serveur du traitement (exp : le prisonnier dépasse la région verte, l'opération de pointage 'vérification' est échouée ou bien l'opération de pointage 'vérification' est réussie).

L'application mobile 'prisonnier' est connectée à un serveur via Internet, elle envoie des notifications au détenu afin de prouver sa présence, donc il doit poser son doigt sur son smartphone pour capturer l'empreinte afin de l'envoyer au serveur qui va la traiter et la comparer avec son empreinte déjà enregistrée dans la base de données.

En cas de conformité de l'empreinte digitale, l'agent master reçoit un message que l'opération de vérification est réussie et lui prolonge un peu de temps avant la prochaine notification.

Si en revanche la conformité n'est pas validé par le serveur il reçoit un message "opération échouée" (même l'agent de police et l'agent master reçoivent ce message), qu'il faut refaire l'opération, le condamné dispose de 2 essais, au bout du 3ème, le serveur envoi une notification à l'*Agent de master et l'agent de police* qui situe dans la même région du condamné pour vérifier la situation de ce dernier (c'est le même processus lorsque le condamné dépasse la région verte). Il existe un 3ème cas, quand le détenu ne répond pas à la notification du pointage (ou bien il clique sur le bouton cancel) ou il dépasse la région verte, une alarme doit déclencher chez l'Agent master et l'agent de police avec la dernière localisation, l'agent de police et l'agent master prennent les mesures nécessaires.

L'avantage de ce modèle par rapport à celui du bracelet électronique c'est que l'application mobile utilise des données biométriques, elle est donc plus fiable que le modèle bracelet comme technologie de surveillance.

Remarque : 1- La vérification de la présence du prisonnier se fait au niveau du serveur, ce dernier traite et compare l'empreinte du pointage avec celle déjà enregistrée dans la base de données par l'utilisation d'un un système de reconnaissance d'empreinte digitale (chapitre 1).

2- L'agent master est le responsable de plusieurs région (exp : bordj,sétif..)

3- Une région contient plusieurs Agents de police.

4- Chaque région est divisée en trois régions : verte, orange et rouge.

5- Chaque prisonnier a une région verte, orange et rouge.

6- Le serveur du traitement détecte l'emplacement du prisonnier 'si ce dernier dépasse la région verte' et émet un 'event' de l'emplacement du prisonnier au serveur websocket, ce dernier envoie une notification de l'emplacement du prisonnier à l'agent de police et l'agent master.

FIGURE 2-2 – Application "SKIA pour le placement sous surveillance mobile"

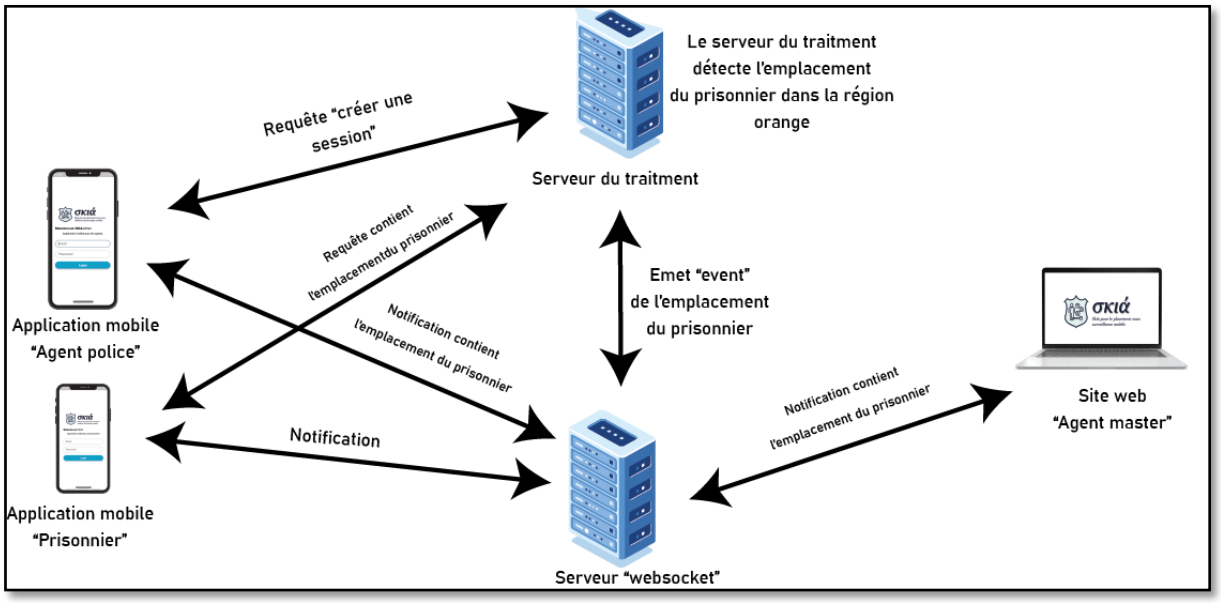


FIGURE 2-3 – Description sommaire du modèle proposé

2.3.1 Le développement de notre application

- Pour l'application web :
  - Nous avons développé **le site web** par le *framework LARAVEL*.

- **Interface** : nous avons créé l'interface de notre application web avec le *jquery* et le *framework bootstrap 5 (HTML + CSS + JS)*.
- **Backend :** nous avons programmé notre application web avec le *PHP version 7.4*.

➢ Pour le serveur de traitement :

- Pour le développement du serveur nous avons utilisé le *framework flask version 2* de *langage python*.

➢ Pour le serveur de websocket :

- Pour le développement du serveur web socket nous avons utilisé la *bibliothèque socket IO* et le *framework nodejs* de langage *javascript*.

➢ Pour les deux applications mobiles:

- Nous avons développé les deux applications mobiles par le *framework flutter* de langage *dart "version 2"*.

➢ Pour le logo :

- Nous avons créé le logo de notre application "SKIA" par le logiciel *Adobe illustrateur cc 2020*.

2.3.2 Architecture informatique du simulateur

1- Application mobile 'Prisonnier'

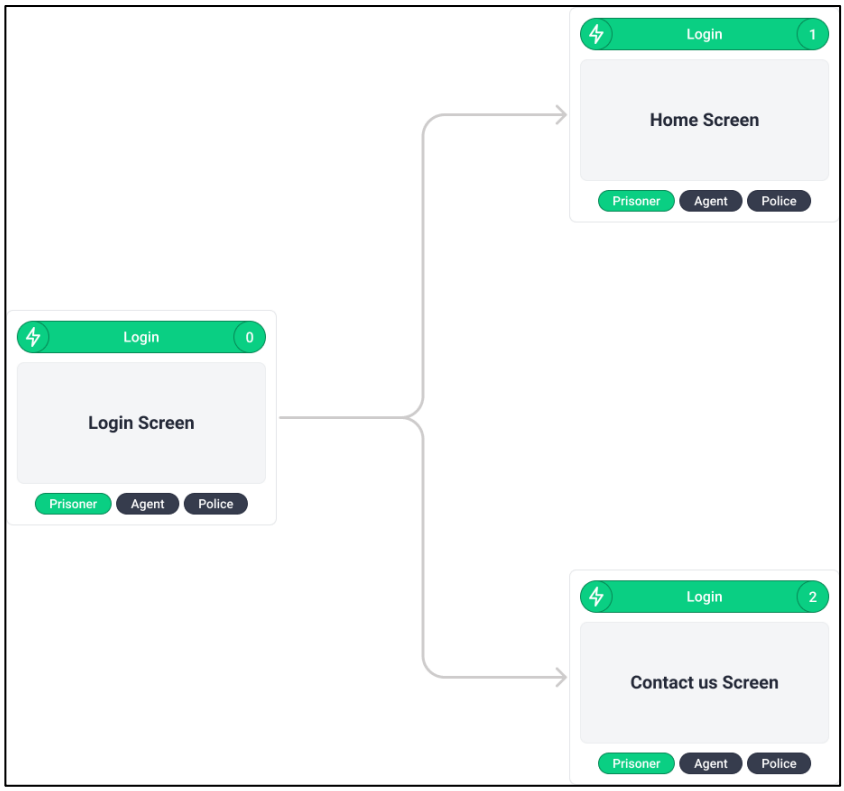


FIGURE 2-4 – ScreenFlow de l'application mobile 'Prisonnier'

2- Application mobile 'Agent de police'

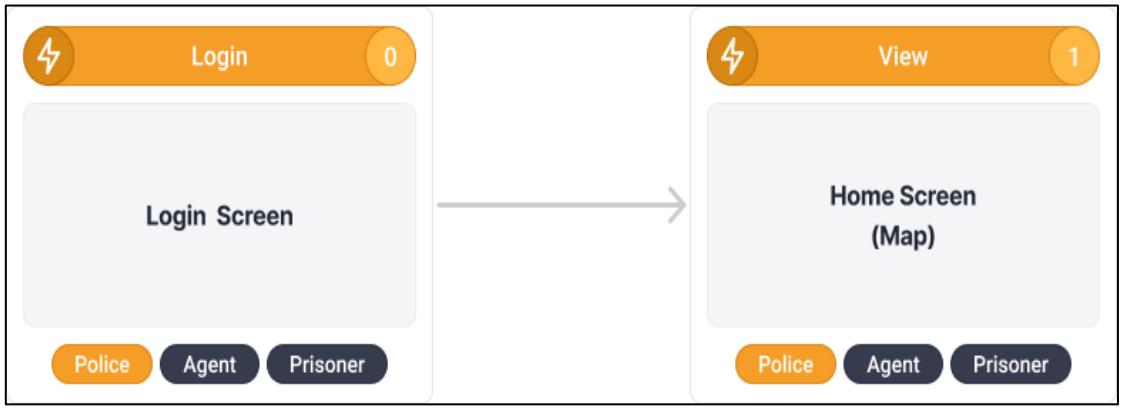


FIGURE 2-5 – ScreenFlow de l'application mobile' Agent de police'

3- Site web 'Agent master'

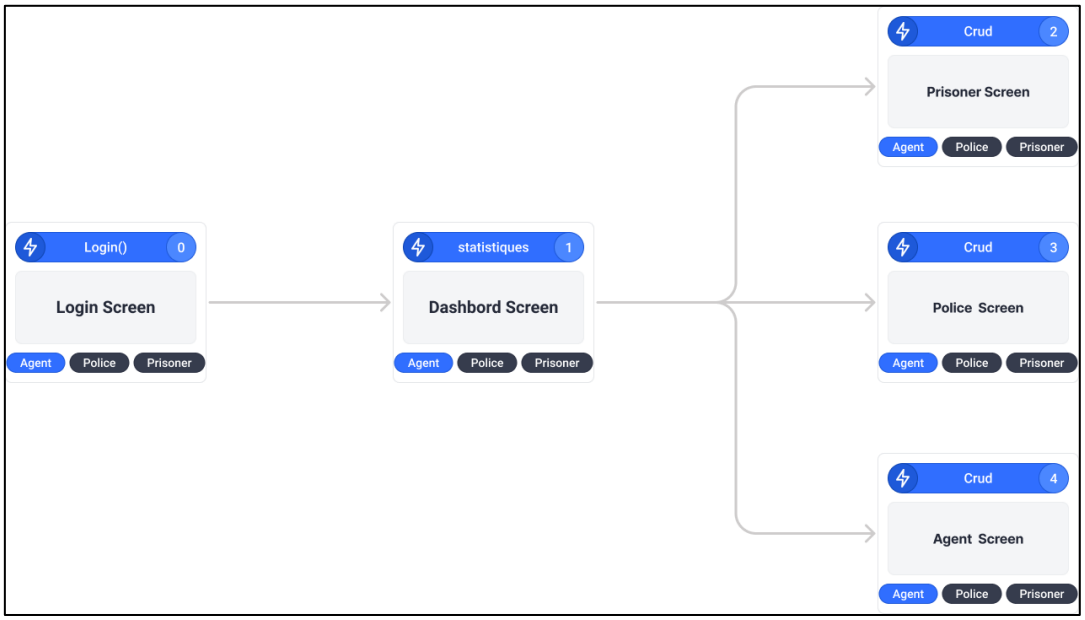


FIGURE 2-6 – ScreenFlow de site web 'Agent master'

2.4 Objectifs

Afin d'améliorer le système de surveillance des détenus, on veut qu'on introduit la *technologie biométrique*, cette dernière peut être la clé pour supprimer les méfaits du bracelet électronique. En effet l'utilisation d'une *application mobile sous Android et iOS* va augmenter considérablement le niveau de suivi et le niveau de sécurité vu qu'elle utilise des paramètres biologiques uniques qui distinguent chaque individu (l'empreinte digitale).

En plus de ce qu'on a mentionné, *l'application mobile sous Android et iOS* est une solution envisageable pour ne pas discriminer les familles des détenus incapables de payer les frais d'installation d'un bracelet électronique. Finalement notre projet est en mesure de respecter la vie des gens, de supprimer l'effet psychologique et d'éviter les mauvais effets sur la santé résultant de l'utilisation des bracelets électroniques (le bracelet électronique fixé à la cheville ne doit jamais être enlevé : le condamné se douche et dort avec).

2.5 Conclusion

Dans ce chapitre, nous avons présenté quelques modèles existants d'application de santé pour malades et professionnel de santé (médecins) de type passeport numérique médical, leurs fonctionnements et leurs objectifs. Après l'analyse de ces modèles, nous avons préparé une description sommaire pour notre propre modèle et nous avons identifié ces principaux objectifs. Dans le chapitre suivant, nous allons entamer la modélisation et la conception de notre modèle.

Chapitre 3

Architecture et modélisation

3.1 Introduction

La réalisation d'un système nécessite la modélisation qui permet d'anticiper, de prévoir et d'étudier les informations relatives à ce système. Pour se faire, on a opté pour le langage UML qui permet de représenter des concepts graphiques et de modéliser les applications. Cette modélisation UML montre les différents acteurs du système ainsi que les rôles qu'ils peuvent tenir.

3.2 Méthodologie de conception

Dans ce qui suit nous allons présenter le langage UML.

3.2.1 Présentation d'UML

UML «Unified Modeling Language» est un langage de modélisation orientée objet développé en réponse à l'appel de la proposition lancée par l'OMG dans le but de définir une notation standard pour la modélisation des applications construites à l'aide d'objets et aussi pour la conception des logiciels. Aussi, UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc [13].

3.2.2 Modèle de conception MVC (Design pattern MVC)

Le pattern MVC permet de bien organiser le code source. Il va nous aider à savoir quels fichiers créer, mais surtout à définir leur rôle. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts [14]:

- ***Modèle :*** cette partie gère les données du site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve donc les requêtes SQL.

Parfois, les données ne sont pas stockées dans une base de données. C'est plus rare, mais on peut être amené à aller chercher des données dans des fichiers. Dans ce cas, le rôle du modèle est de faire les opérations d'ouverture, de lecture et d'écriture de fichiers.

- ***Vue* :** cette partie se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple la liste des messages des forums.

- ***Contrôleur* :** cette partie gère la logique du code qui prend des décisions. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le contrôleur contient exclusivement du PHP. C'est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d'accès).

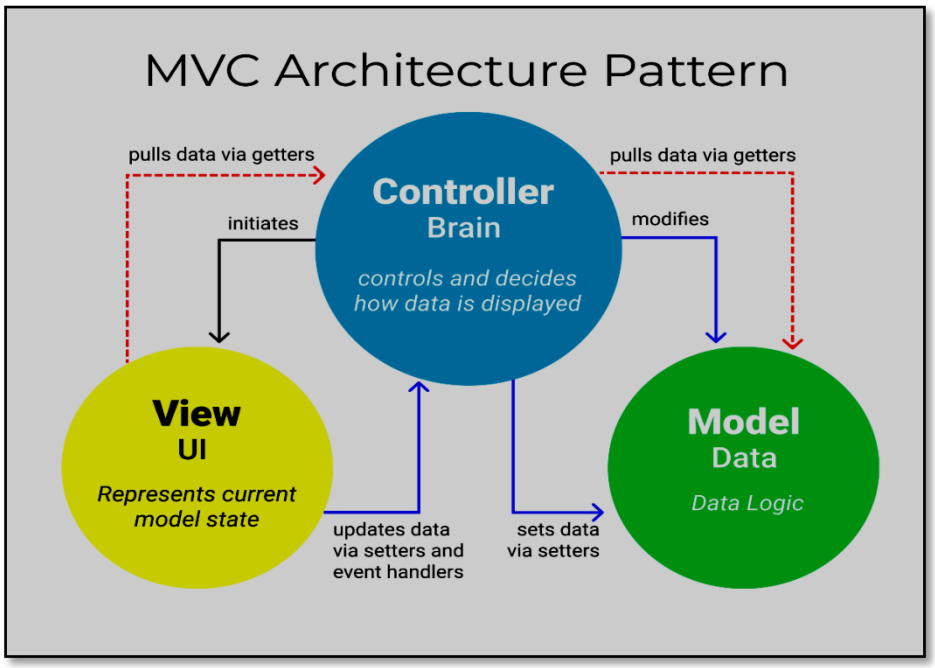


FIGURE 3-1- Diagramme explicatif de MVC

3.2.3 JSON Web Token (JWT)

JSON Web Token (JWT) est un moyen compact et sûr pour les URL de représenter les créances à transférer entre deux parties. Les revendications d'un JWT sont codées sous la forme d'un objet JSON utilisé comme charge utile d'une structure JSON Web Signature (JWS) ou comme texte en clair d'une structure JSON Web Encryptions (JWE), ce qui permet de signer numériquement les revendications ou d'en protéger l'intégrité par un code d'authentification de message (MAC) et/ou de les crypter [15].

Un jeton Web JSON ressemble à ceci (les retours à la ligne sont insérés pour faciliter la lecture) :

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.
TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

Ce jeton représente cette structure de JSON :

{
"alg":"HS256",
"typ":"JWT"
}
"sub" :"1234567890",
"name":"Hichem Merniz",
"admin":true
}

3.3 Analyse et conception

3.3.1 Diagramme de cas d'utilisation

Le Diagramme de cas d'utilisation est utilisé pour la modélisation des besoins des utilisateurs. Les cas d'utilisations décrivent le comportement du système étudié du point de vue de l'utilisateur, et les possibilités d'interactions fonctionnelles entre le système et les acteurs, ils permettent de définir les limites et les relations entre le système et son environnement [13].

3.3.1.1 Rôle du diagramme de cas d'utilisation

- Donne une vue du système dans son environnement extérieur.
- Définit la relation entre l'utilisateur et les éléments que le système met en œuvre.

3.3.1.2 Les composants d'un diagramme de cas d'utilisation

Les composants de base des diagrammes de cas d'utilisation sont l'acteur, le cas d'utilisation, et l'association [13].

- **Acteur :** Un acteur est un utilisateur qui communique et interagit avec les cas d'utilisation du système. C'est une entité ayant un comportement comme une personne ou système.

- **Cas d'utilisation :** Un cas d'utilisation représente une fonctionnalité fournie par le système, typiquement décrite sous la forme Verbe+objet (par exemple immatriculer voiture, effacer utilisateur). Les cas d'utilisation sont représentés par une ellipse contenant leurs noms.

- **Association :** Les associations sont utilisées pour lier des acteurs avec des cas d'utilisation. Elles indiquent qu'un acteur participe au cas d'utilisation sous une forme quelconque. Les associations sont représentées par une ligne reliant l'acteur et le cas d'utilisation.

3.3.1.3 Diagrammes de cas d'utilisation de notre simulateur

Le diagramme de cas d'utilisations de notre application mobile 'Prisonnier' est modulé comme suit :

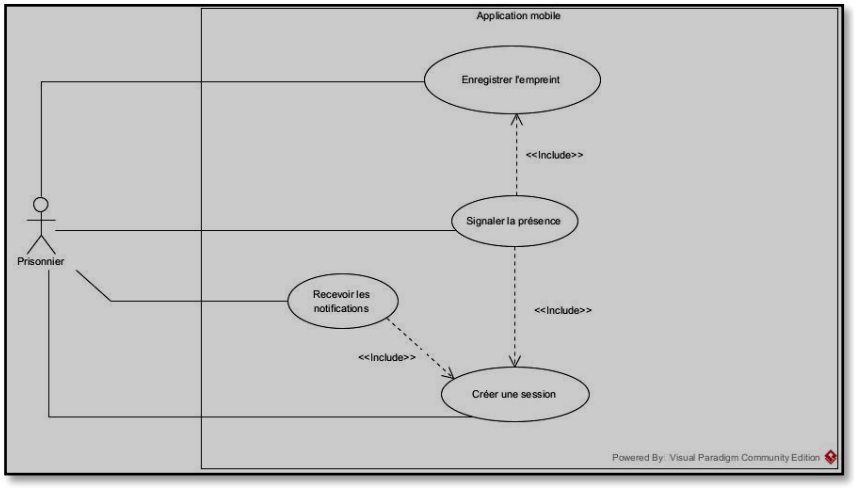


FIGURE 3-2- Diagramme de cas d'utilisation d'Application mobile 'Prisonnier'

Remarque : le cas d'utilisation 'Créer une session' c'est le cas d'utilisation 'Authentifier'.

- ✓ L'authentification du prisonnier fonctionne comme suit :
  - le serveur génère un jeton [16] qui certifie l'identité de l'utilisateur puis il l'envoie au client (application mobile). Le client renverra le jeton au serveur pour chaque demande ultérieure de sorte que le serveur sache que la demande provient d'une identité particulière (**le prisonnier**).

Le diagramme de cas d'utilisations de notre application mobile 'Agent de police' est modulé comme suit :

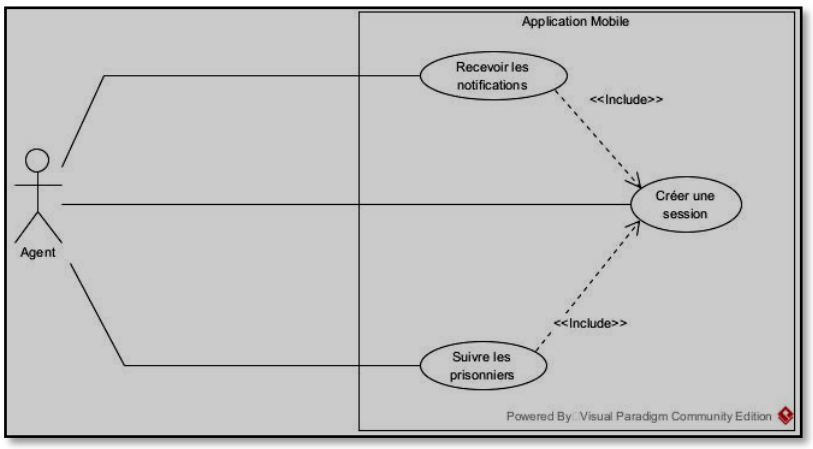


FIGURE 3-3- Diagramme de cas d'utilisation générale d'Application mobile 'Agent de police'

Remarque : le cas d'utilisation 'Créer une session' c'est le cas d'utilisation 'Authentifier'.

Le diagramme de cas d'utilisations de notre site web 'Agent master' est modulé comme suit :

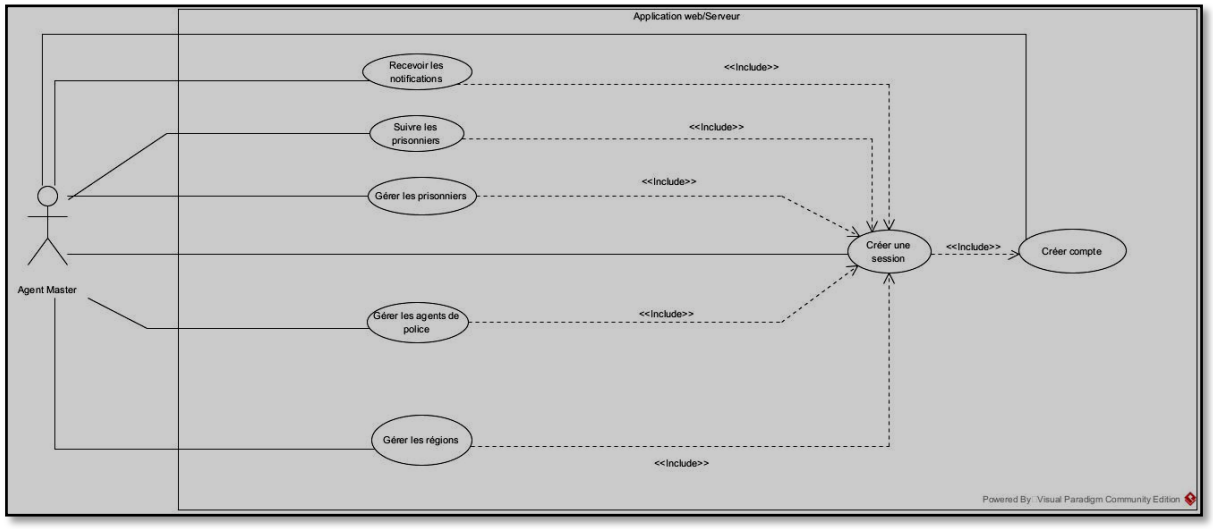


FIGURE 3-4- Diagramme de cas d'utilisation générale 'Site web Agent master'

Remarque : le cas d'utilisation 'Créer une session' c'est le cas d'utilisation 'Authentifier'.

3.3.1.4 Description textuelle des cas d'utilisation

Un cas d'utilisation 'CU' permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié [17].

- **Pré condition :** définissent les conditions qui doivent être satisfaites pour que la CU puisse démarrer.
- **Post condition :** définissent ce qui doit être vrai lorsque le CU se termine avec succès, qu'il s'agisse d'un scénario nominal ou alternatif.

3.3.1.4.1 Cas d'utilisation « Créer une session »

Ce tableau illustre le cas d'utilisation de '*Créer une session*' ou bien 'Authentifier' et présente les différents acteurs qui ont accès à ce service.

TABLEAU 3-1- Créer une session 'Authentifier'

Identification • **Nom du cas d'utilisation :** Créer une session. • **But :** Avoir un accès à l'application mobile ou web. • **Acteur :** Prisonnier, Agent de police et Agent master
Séquencement • **Précondition :** Créer un compte. • **Enchainements nominaux :** - L'utilisateur saisit son username et son mot de passe. • **Enchainements alternatifs :** - Username saisis non valides, - Mot de passe non valide. • **Post-conditions :** - L'utilisateur accède au système.

3.3.1.4.2 Cas d'utilisation « Signaler la présence d'un prisonnier»

Ce tableau illustre le cas d'utilisation de '*Signaler la présence d'un prisonnier*' et présente les différents acteurs qui ont accès à ce service.

TABLEAU 3-2-Signaler la présence d'un prisonnier

Identification • **Nom du cas d'utilisation :** Signaler la présence d'un prisonnier. • **But :** Activer la présence du prisonnier & contrôler si effectivement un condamné est chez lui ou pas par l'utilisation de l'empreinte digitale. • **Acteur :** Prisonnier.
Séquencement • **Précondition :** S'authentifie (créer une session) • **Enchainements nominaux :**

- L'utilisateur reçoit une notification 'Scan your fingerprint to authenticate'; - L'utilisateur pose son doit sur son smartphone pour la lecture et la vérification d'empreinte. - L'application envoie l'empreinte au serveur du traitement. • **Enchainements alternatifs :** - Les informations biométriques (empreinte) non valide. - L'utilisateur dépasse l'intervalle du temps du pointage. - L'utilisateur clique sur le bouton cancel du pointage • **Post-conditions :** - Le serveur du traitement traite et comparer l'empreinte du pointage avec celle déjà enregistrer dans la base de données.

3.3.1.4.3 Cas d'utilisation « Ajouter prisonnier»

Ce tableau illustre le cas d'utilisation de '*Ajouter prisonnier*' et présente les différents acteurs qui ont accès à ce service.

TABLEAU 3-3-Ajouter prisonnier

Identification • **Nom du cas d'utilisation :** Ajouter prisonnier. • **But :** Ajouter un nouveau prisonnier au système • **Acteur :** Agent master (Administrateur).
Séquencement • **Précondition :** S'authentifie. • **Enchainements nominaux :** - L'utilisateur saisit les informations du prisonnier - L'utilisateur enregistre l'empreinte du prisonnier. - L'utilisateur enregistre le nouveau prisonnier dans la base de données

- **Enchainements alternatifs :**

- Données saisies non valides.

- L'oubli d'un champ.

- Les informations biométriques (empreinte) non valide.

- **Post-conditions :**

- Mise à jours de la base de données.

- La création d'un nouveau compte 'prisonnier'.

3.3.1.4.4 Cas d'utilisation « Suivre prisonnier – vérifier sa présence et son emplacement-»

Ce tableau illustre le cas d'utilisation de '*Suivre prisonnier-vérifier sa présence et son emplacement-'* et présente les différents acteurs qui ont accès à ce service.

TABLEAU 3-4-Suivre prisonnier- vérifier sa présence et son emplacement-

Identification

- **Nom du cas d'utilisation :** Suivre prisonnier - vérifier sa présence et son emplacement-
- **But :** Vérifier la présence et l'emplacement du prisonnier.
- **Acteur :** Agent master (Administrateur).

Séquencement

- **Précondition :**

- S'authentifie.

- Le pointage du prisonnier.

- **Enchainements nominaux :**

-L'application mobile 'Prisonnier' envoie l'empreinte du pointage et l'emplacement (en temps réel) du prisonnier au serveur du traitement.

- Le serveur traite et compare l'empreinte du pointage avec celle déjà enregistrée dans la base de données (le traitement et la comparaison se font par un système de reconnaissance d'empreinte digitale (chapitre1)) et calcule l'emplacement du prisonnier.

- Le serveur envoie le résultat de la vérification à l'agent master (si la vérification réussie) ou bien à l'agent master et l'agent de police (si la vérification échouée) et émet un 'event' de l'emplacement du prisonnier au serveur websocket, ce dernier envoie une notification de l'emplacement du prisonnier à l'agent master (si le condamné est dans la région verte) et à l'agent de police, l'agent master et prisonnier (si le condamné dépasse la région verte).

- **Enchainements alternatifs :**

- L'empreinte de pointage non valide.

- Le prisonnier clique sur le bouton cancel du pointage.

- **Post-conditions :**

- Consulter la liste des prisonniers présents et absents.

- L'agent de police et l'agent master prennent les mesures nécessaires si la vérification est échouée ou bien le condamné dépasse la région verte.

3.3.2 Diagramme de séquence

Les diagrammes de séquence représentent des échanges de messages entre objets. Ils doivent rester aussi simples que possible et seuls les messages pertinents doivent être représentés [18].

- ✓ L'axe vertical représente le temps.
- ✓ L'axe horizontal représente les objets/acteurs impliqués dans l'interaction.
- ✓ Une ligne verticale est attachée à chaque objet/acteur et représente sa ligne de vie «lifeline».

1- Les objets/acteurs

Sur un diagramme de séquence, les objets apparaissent toujours dans la partie supérieure, ce qui facilite l'identification des classes qui participent à l'interaction.

2- Les messages

Les messages sont représentés par des flèches directionnelles.

Cette représentation est similaire à celle d'une association sur un diagramme de classes cependant, les messages servent à représenter la communication entre les objets, et non la relation structurelle présente entre les classes. Au-dessus des flèches directionnelles figure un texte informant du message envoyé entre les objets.

3- La ligne de vie de l'objet/acteur

Ce concept représente la vie d'un objet dans le contexte de la séquence d'événements. Les objets qui sont créés vers la fin de la séquence n'apparaissent pas toujours en haut du diagramme, mais peuvent apparaître à l'endroit où ils sont créés.

4- Le point de contrôle (barre d'activation)

Ce concept illustre la période pendant laquelle un objet effectue une action.

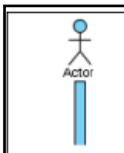
	Acteur	Les acteurs peuvent communiquer avec des objets, ainsi ils peuvent eux aussi être énumérés en colonne. Un acteur est modélisé en utilisant le symbole habituel: Stickman.
	Objet	Les objets sont des entités appartenant au système (instance d'une classe) ou se trouvant à ses limites (acteurs)
	Ligne de vie	Elle est représentée par une ligne verticale en dessous des objets, représente la période de temps durant laquelle l'objet "existe".
	Message récursif	L'envoi de messages récursifs se représente par un dédoublement de la bande d'activation
	Message	Les objets communiquent en échangeant des messages représentés sous forme de flèches, ils sont étiquetés par le nom de l'opération ou du signal invoqué.
	Message de retour	Représenté par une flèche discontinue, c'est la réponse au message envoyé.

FIGURE 3-5- Représentation d'un diagramme de séquence.

3.3.2.1 Diagramme de séquence « Créer une session 'Authentification' Agent master »

L'authentification consiste à assurer la confidentialité des données, elle se base sur la vérification du login et du mot de passe. Ces informations sont préétablies dans une base de données. Lors de l'authentification de l'utilisateur, deux cas peuvent se présenter : informations correctes ou incorrectes, ce qui explique l'utilisation de l'opérateur « alt ». Si les informations fournies sont correctes, alors le système accorde l'accès à l'interface appropriée. En revanche, si l'utilisateur saisit des informations incorrectes, le système génère un message d'erreur et réaffiche la page d'authentification d'où l'utilisation de l'opérateur «loop».

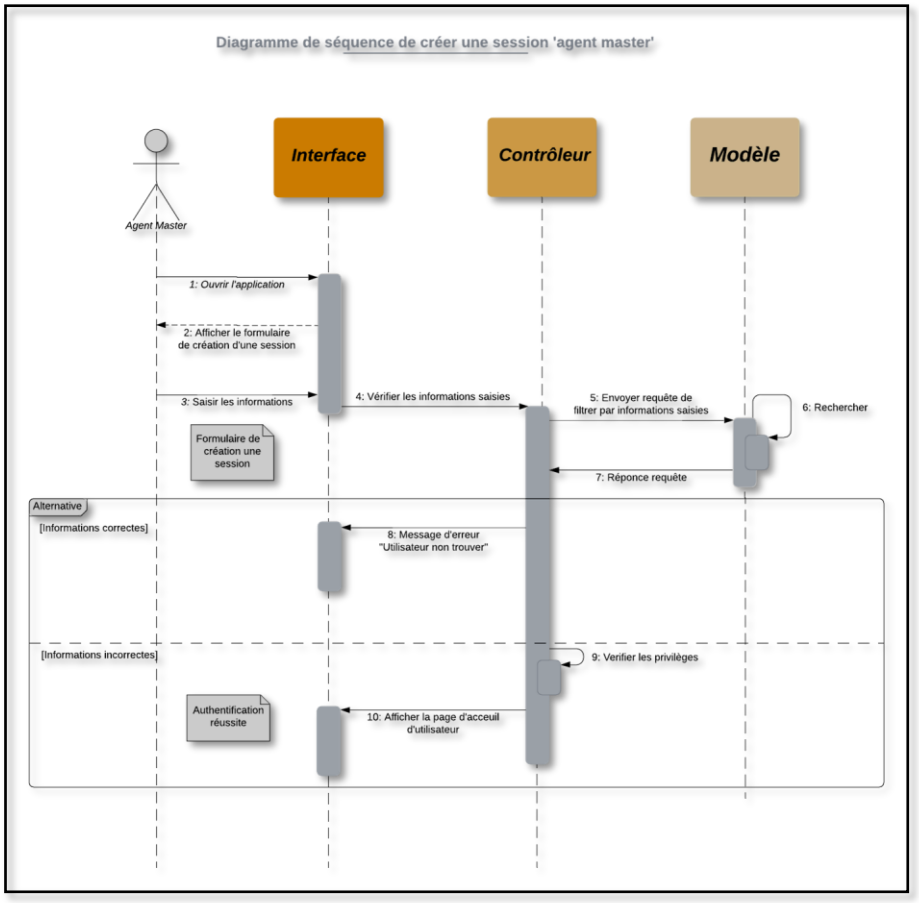


FIGURE 3-6- Diagramme de séquence de création d'une session 'Authentification' -Agent master'

3.3.2.2 Diagramme de séquence « Ajouter un prisonnier »

Le diagramme de séquence suivant illustre les interactions nécessaires pour ajouter un prisonnier, l'utilisateur (Agent master) saisit les champs nécessaires du formulaire "l'ajout d'un prisonnier". La création du prisonnier se fait après une vérification des champs, finalement le site web affiche le succès d'ajout ou bien une erreur de validation des champs.

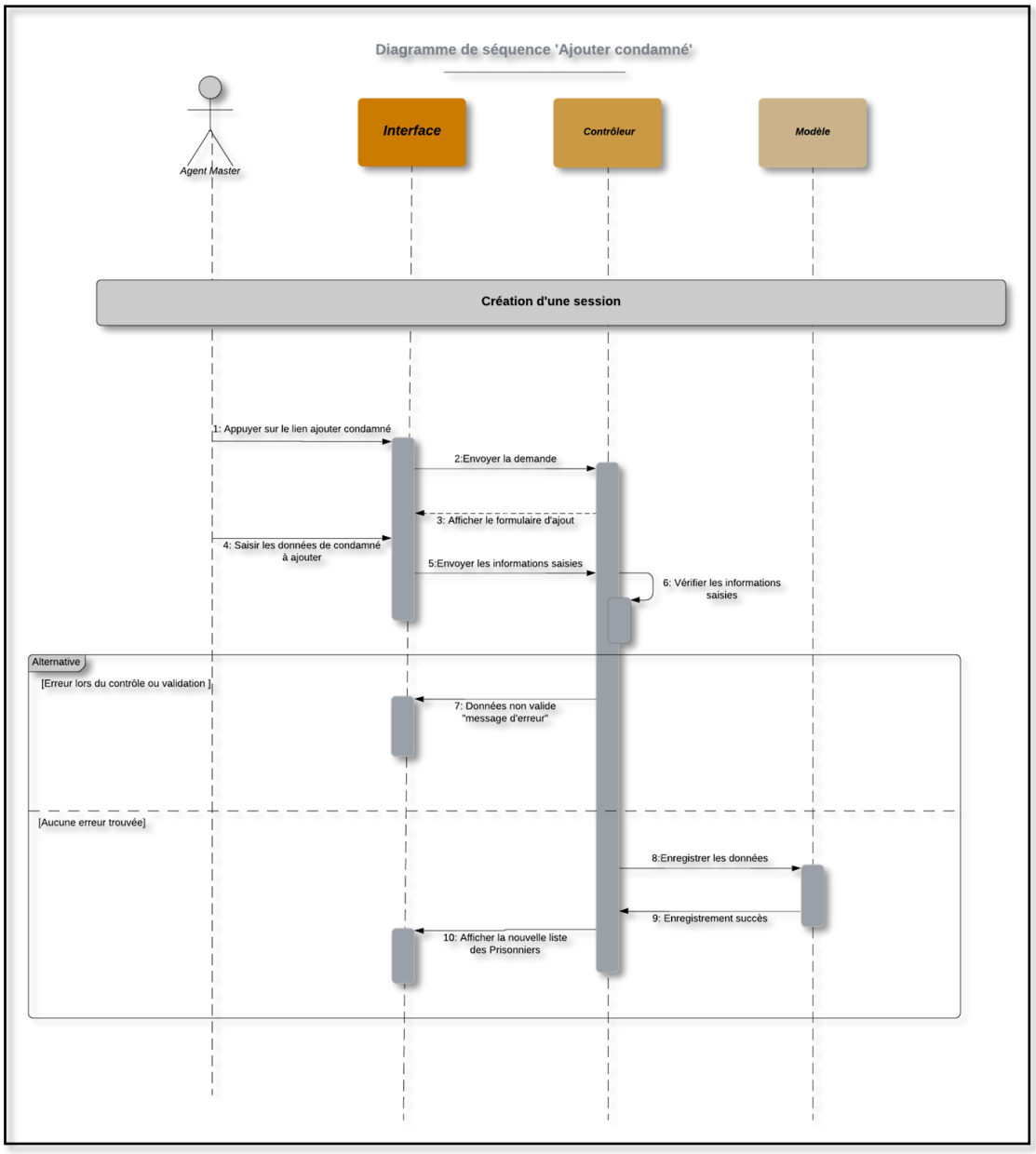


FIGURE 3-7- Diagramme de séquence d'Ajouter un prisonnier

3.3.2.3 Diagramme de séquence « Créer une session 'Prisonnier' -Authentifier- »

Le serveur génère un jeton [16] qui certifie l'identité de l'utilisateur puis il l'envoie au client (application mobile). Le client renverra le jeton au serveur pour chaque demande ultérieure de sorte que le serveur sache que la demande provient d'une identité particulière (**le prisonnier**).

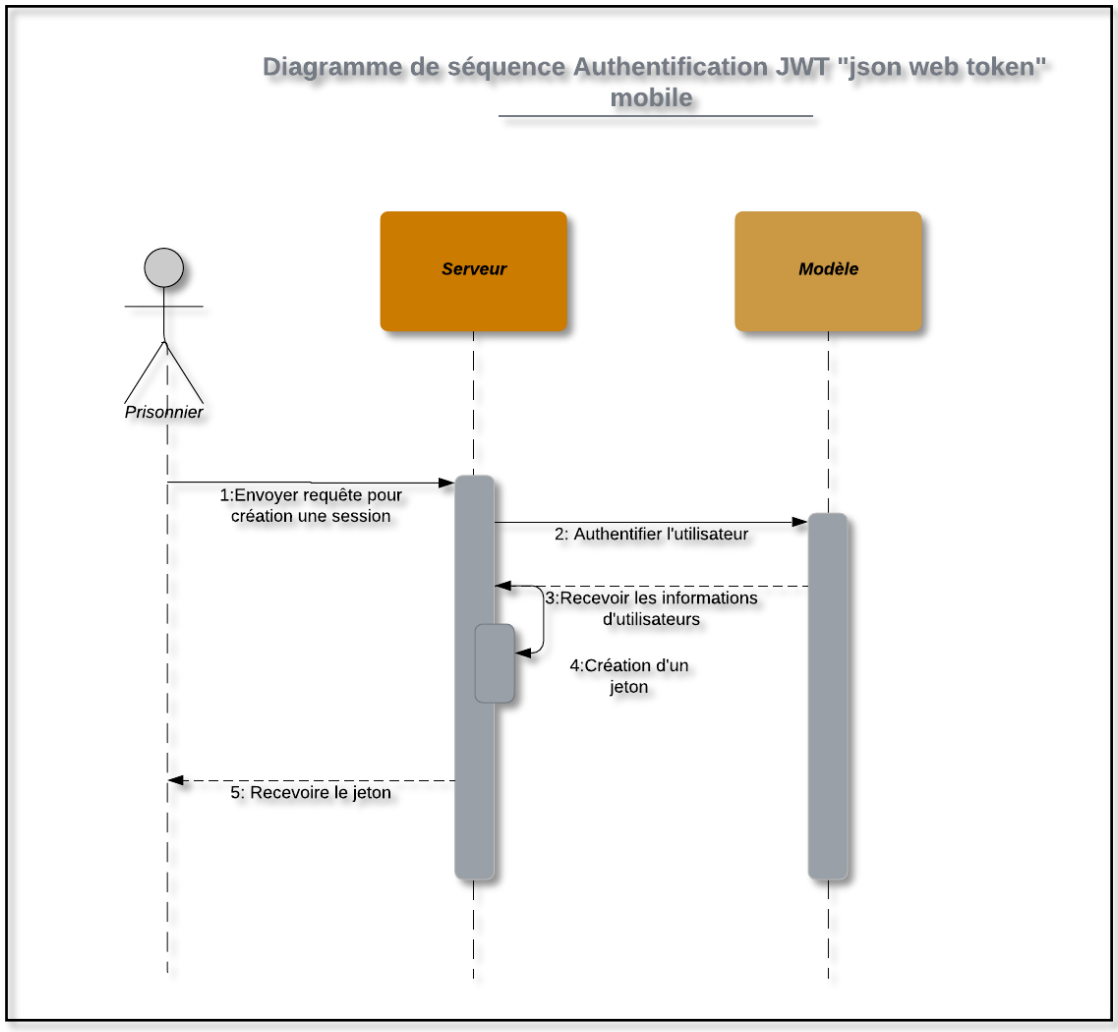


FIGURE 3-8- Diagramme de séquence de Créer une session 'Prisonnier' -Authentifier-

3.3.2.4 Diagramme de séquence « Signaler présence 'Prisonnier'»

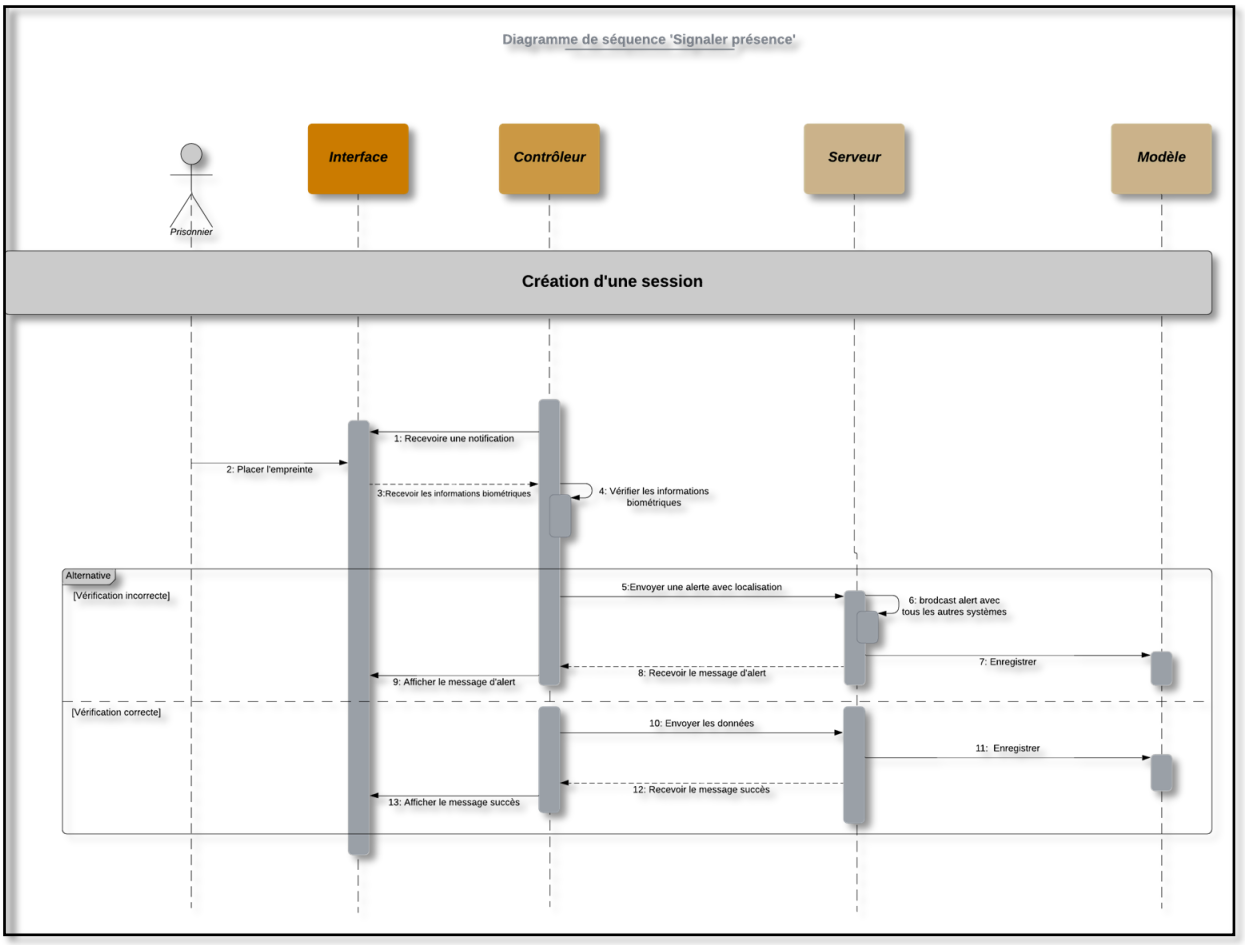


FIGURE 3-9- Diagramme de séquence de Signaler présence 'Prisonnier'

3.3.3 Diagramme de classe

Après l'étude détaillée des cas d'utilisation, nous avons déduit le diagramme de classe global du système. Ce diagramme est considéré comme la phase finale de la conception théorique de notre système et sera pris comme la référence à partir de laquelle va se dérouler le développement logiciel, et l'écriture du code source de notre application.

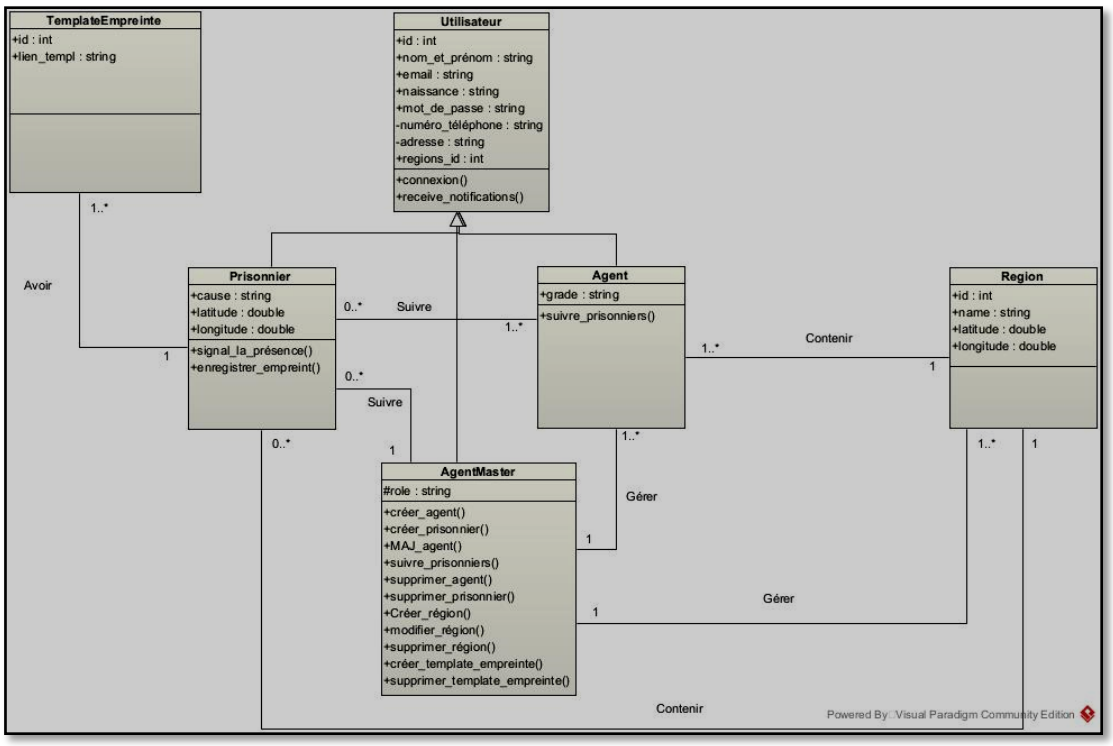


FIGURE 3-10- Diagramme de classe de l'application 'SKIA pour le placement sous surveillance mobile'

3.4 Conclusion

Dans ce chapitre, nous avons traité la phase d'analyse et conception qui est faite grâce à des diagrammes UML : diagrammes de cas d'utilisation, de séquences et de classes.

Chapitre 4

Implémentation

4.1 Introduction

Pour pouvoir mener à bien un projet informatique, il est nécessaire de choisir des technologies permettant de simplifier sa réalisation. Pour cela, après avoir compléter le contenu du chapitre précédant 'Architecture et modélisation', nous aborderons la partie implémentation dans ce qui suit.

Dans ce chapitre nous présenterons la description des environnements matériels et logiciels qui nous ont permis de réaliser notre projet, des technologies et des langages de programmation que nous avons utilisée. Ensuite nous expliquerons le fonctionnement de notre simulateur 'SKIA pour le placement sous surveillance mobile' en présentant ses différentes interfaces qui permettent l'interaction entre l'utilisateur et le système.

4.2 Environnement du travail

4.2.1 Environnement matériel

Pour la réalisation de notre projet, nous avons utilisé un ordinateur toshiba satellite caractérisé par :

- Système d'exploitation : Windows10.
- Processeur : AMD A6-6310 APU with AMD Redeon R4 Graphics 1.80 GHz
- Mémoire vive : 8 Go.
- Disque Dur : 1 To.

Pour les différentes étapes de test, d'installation et de déploiement de l'application nous avons eu besoin d'une terminale mobile supportant le système d'exploitation Android dont les caractéristiques sont les suivantes :

- Nom de l'appareil : Samsung M20
- Système d'exploitation : Android 10.
- Connexion : 4G, ADSL.
- Mémoire vive : 2 GB.
- Disque Dur : 16 GB.

4.2.2 Environnement logiciel

4.2.2.1 Visual Studio

Visual Studio Code est un éditeur de code source léger et puissant, disponible sur toutes les plateformes (Windows, MacOs et Linux). Il est livré avec un support intégré pour JavaScript, Type Script et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages (tels que C++, C#, Java, Python, PHP, Go) et les moteurs d'exécutions tels que (.NET et Unity) [19].

4.2.2.2 Windows terminal

Le terminal Windows est une application moderne, rapide, efficace, puissante et productive destinée aux utilisateurs d'outils de ligne de commandes et de Shell tels que Command Prompt, PowerShell et WSL. Ses principales caractéristiques sont les suivantes : onglets multiples, volets, prise en charge des caractères Unicode et UTF-8, moteur de rendu de texte accéléré par le GPU, thèmes, styles et configurations personnalisés [20].

4.2.2.3 Adobe Illustrator

C'est une application d'illustration vectorielle de référence pour toutes les créations, des visuels pour le web et les appareils mobiles aux logos, icônes, illustrations éditoriales et design de packaging en passant par les panneaux publicitaires [21].

4.2.2.4 Postman

Postman est une plateforme d'API permettant de créer et d'utiliser des API. Postman simplifie chaque étape du cycle de vie des API et rationalise la collaboration afin que vous puissiez créer de meilleures API, plus rapidement [22].

4.2.2.5 Xampp

XAMPP est une distribution Apache entièrement gratuite et facile à installer contenant MySQL, PHP et Perl. Le paquetage open source XAMPP a été mis au point pour être incroyablement facile à installer et à utiliser [23].

4.2.2.6 Dart

Dart est un langage optimisé pour le client permettant de développer des applications rapides sur n'importe quelle plateforme. Son objectif est d'offrir le langage de programmation le plus productif pour le développement multiplateformes, associé à une plateforme d'exécution flexible pour les Framework d'applications [24].

4.2.2.7 Python

Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est interprété, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes [25].

4.2.2.8 JavaScript (JS)

JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page web. À chaque fois qu'une page web fait plus que simplement afficher du contenu statique — afficher du contenu mis à jour à des temps déterminés, des cartes interactives, des animations 2D/3D, des menus vidéo défilants, etc...

JavaScript a de bonnes chances d'être impliqué. C'est la troisième couche des technologies standards du web, les deux premières (HTML et CSS) étant couvertes bien plus en détail dans d'autres tutoriels sur MDN [26].

4.2.2.9 Php

PHP est un langage de script populaire et polyvalent, particulièrement adapté au développement Web. Il est rapide, flexible et pragmatique, le PHP alimente tout, de votre blog aux sites web les plus populaires du monde [27].

4.2.2.10 Flutter

Flutter est un SDK mobile open-source que les développeurs peuvent utiliser pour créer des applications Android et iOS d'apparence native à partir de la même base de code. Flutter existe depuis 2015, date à laquelle Google l'a présenté, et est resté en phase bêta avant son lancement officiel en décembre 2018 [28].

4.2.2.11 Flask

Flask est un cadre d'application web écrit en Python. Il a été développé par Armin Ronacher, qui dirigeait une équipe internationale de passionnés de Python appelée Poocco. Flask est basé sur la boîte à outils WSGI Werkzeg et le moteur de Template Jinja2 [29].

4.2.2.12 Node js

Node.js est un moteur d'exécution JavaScript piloté par les événements. Node a une myriade d'utilisations potentielles pour le développement JavaScript, y compris un excellent environnement pour construire des applications réseau efficaces [30].

4.2.2.13 Bootstrap

Bootstrap est une collection géante de morceaux de code pratiques et réutilisables écrits en HTML, CSS et JavaScript. Il s'agit également d'un cadre de développement frontal qui permet aux développeurs et aux concepteurs de créer rapidement des sites Web entièrement réactifs [31].

4.2.2.14 Laravel

LARAVEL est un Framework du langage de programmation PHP avec une syntaxe expressive et élégante. Un Framework web fournit une structure et un point de départ pour la création de votre application, ce qui vous permet de vous concentrer sur la création de quelque chose d'extraordinaire pendant que nous nous occupons des détails [32].

4.3 Présentation des interfaces de notre simulateur

Les interfaces graphiques de l'application sont très importantes, car elles permettent de faciliter le dialogue entre l'homme et la machine ainsi que d'améliorer les performances de l'application.

Dans cette partie nous présentons les principales fonctionnalités de notre simulateur par la description de quelques interfaces.

4.3.1 Interface Splash 'Logo de notre simulateur 'SKIA' '

La FIGURE 4.1 illustre l'interface du logo de l'application. Cette interface dure trois secondes au maximum.

FIGURE 4-1- Interface Splash 'Logo de notre simulateur 'SKIA pour le placement sous surveillance mobile'

4.3.2 Interface 'Admin Login ' Application web –Agent master-

Si l'agent master possède déjà un compte, il saisit juste correctement son Email et son Password, ensuite il clique sur '**Login**'.

Cette interface 'login' permet à l'agent master de créer une session- d'authentifier- afin d'accéder à l'interface Accueil.

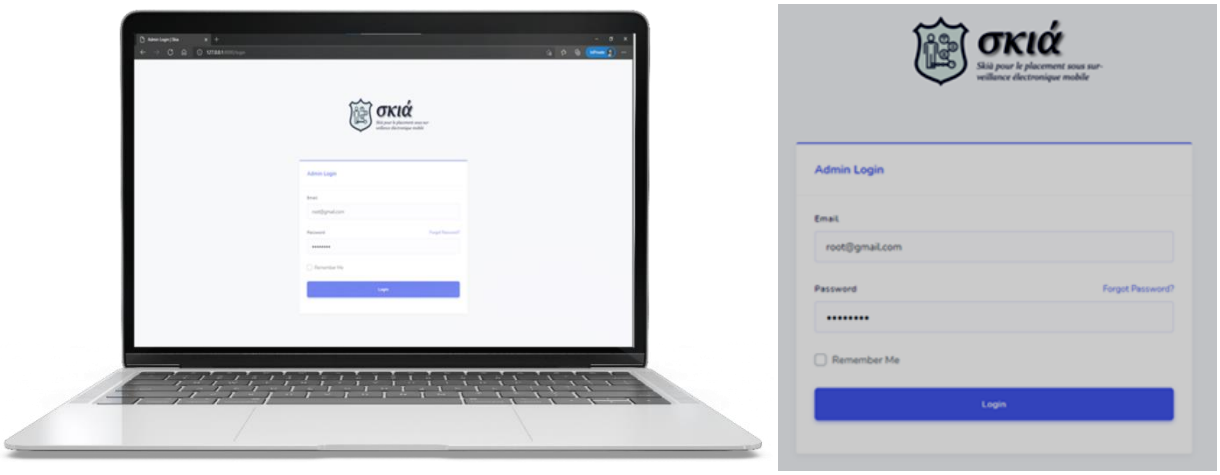


FIGURE 4.2- Interface 'Admin Login' Application web –Agent master-

4.3.3 Interface 'New Region '

Cette interface permet à l'agent master d'ajouter une nouvelle région afin de l'affecter à un prisonnier ou bien à un agent de police.

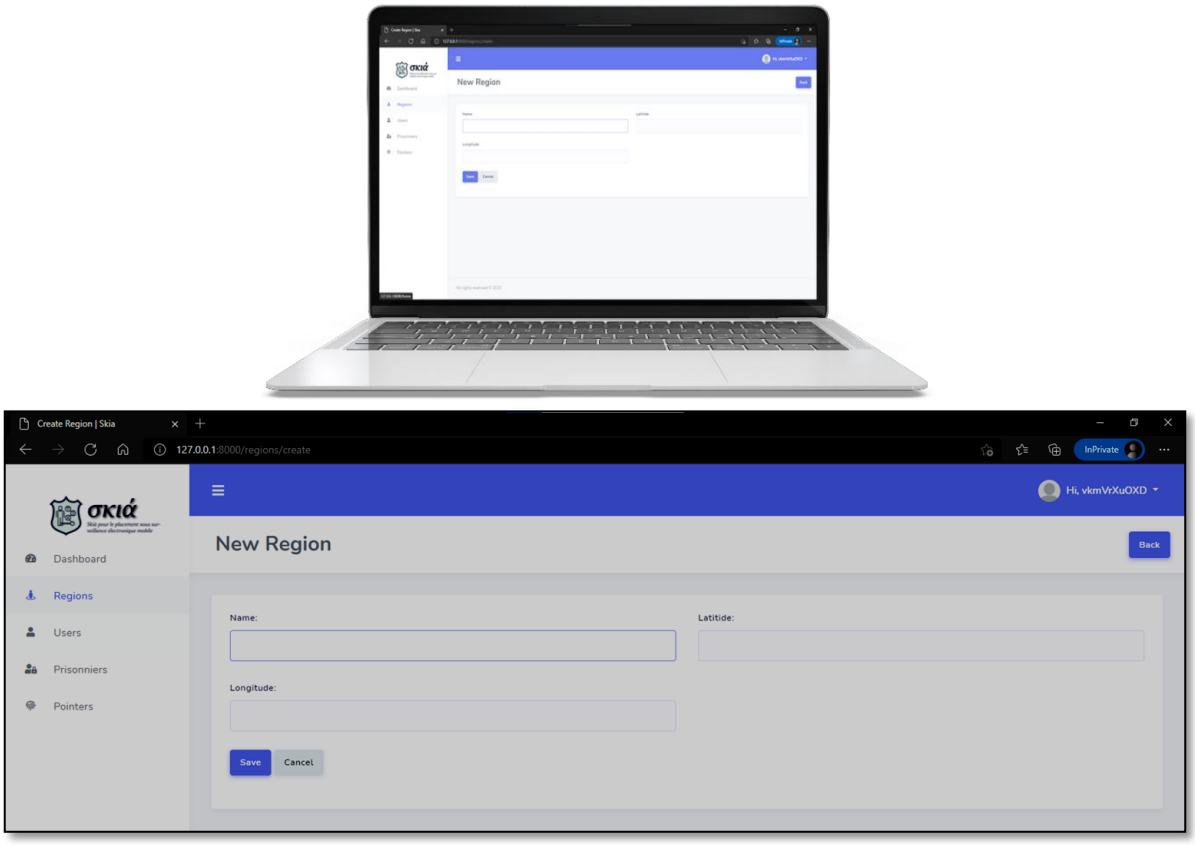


FIGURE 4-3- Interface 'New Region'

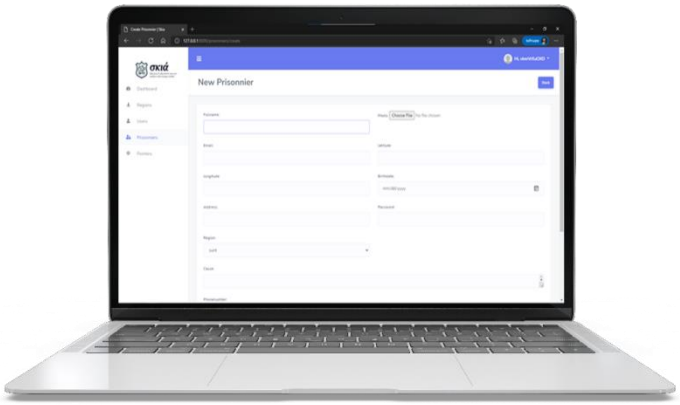
Le code suivant explique le fonctionnement de l"ajout d'une nouvelle région:

```
51      /**
52       * Store a newly created Region in storage.
53       *
54       * @param CreateRegionRequest $request
55       *
56       * @return Response
57       */
58      public function store(CreateRegionRequest $request)
59      {
60          $input = $request->all();
61
62          $region = $this->regionRepository->create($input);
63
64          Flash::success('Region saved successfully.');
65
66          return redirect(route('regions.index'));
67      }
68
```

FIGURE 4-4- Code de l'ajout d'une nouvelle région

4.3.4 Interface 'New Prisonnier '

Cette interface permet d'ajouter un nouveau prisonnier à une région.



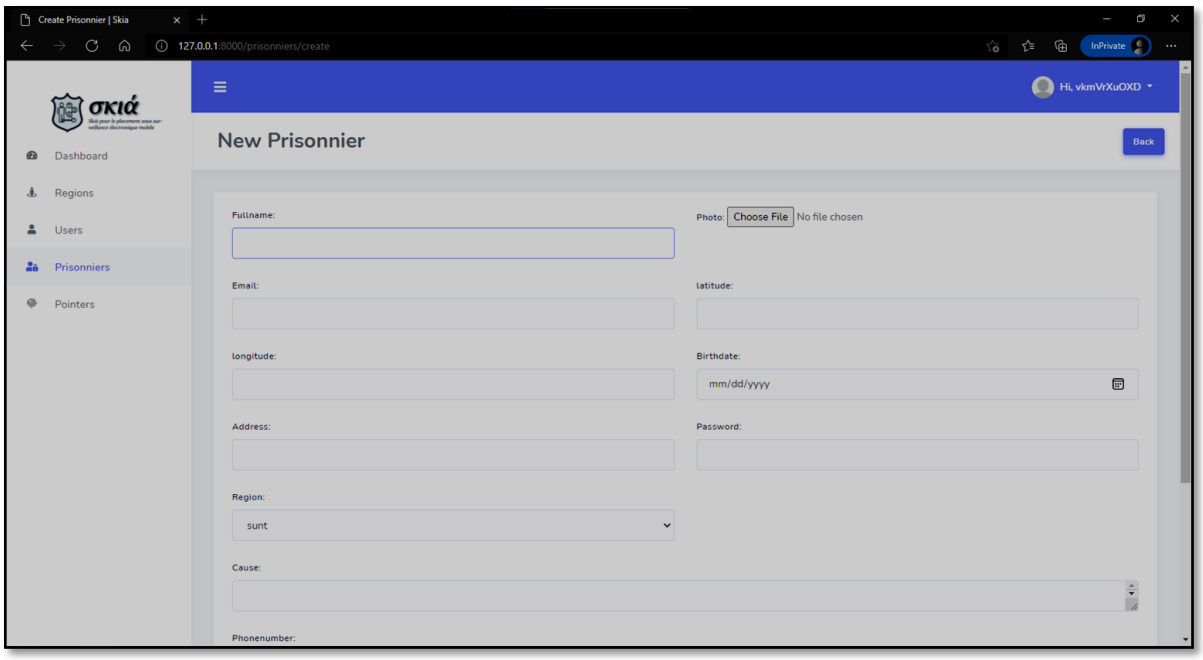


FIGURE 4-5- Interface 'New Prisonnier'

Le code suivant explique le fonctionnement de l'ajout d'un nouveau prisonnier :

```php
public function store(CreatePrisonnierRequest $request)
{
    $input = $request->all();
    if ($request->photo != null) {
        $image_64 = $request->file('photo');
        $replace = substr($image_64, 0, strpos($image_64, ',') + 1);
        $image = str_replace($replace, '', $image_64);
        $image = str_replace(' ', '+', $image);
        $imageName = Str::random(10) . '.' .'jpg';
            $fileNameToStore = Str::random(10) . '.' . 'jpg';
            $resize = \Image::make($image_64)->resize(600, 600, function ($constraint) {
                $constraint->aspectRatio();
              })->encode('jpg');
            $hash = md5($image_64->__toString());
            $image = $hash . "jpg";
            $save = Storage::put("public/user/thumbnails/{$fileNameToStore}", $resize->__toString());
            $url = asset('/storage/user/thumbnails/' . $fileNameToStore);
        } else {
        $url = asset('/storage/noData/noData.png');
    }

        $prisonnier = new Prisonnier;
        $prisonnier->fullName = $request->fullName;
        $prisonnier->photo = $url;
        $prisonnier->email = $request->email;
        $prisonnier->latitude = $request->latitude;
        $prisonnier->longitude = $request->longitude;
        $prisonnier->birthDate = $request->birthDate;
        $prisonnier->address = $request->address;
        $prisonnier->region_id = $request->region_id;
        $prisonnier->cause = $request->cause;
        $prisonnier->password = bcrypt($request->password);
        $prisonnier->phoneNumber= $request->phoneNumber;
        $prisonnier->save();

        Flash::success('Prisonnier saved successfully.');

        return redirect(route('prisonniers.index'));
    }
```

FIGURE 4-6- Code de l'ajout d'un nouveau prisonnier

Afin de compléter l'étape de l'ajout d'un nouveau prisonnier, ce dernier pose ses doigts sur son smartphone afin d'enregistrer ces empreintes dans la base de donnée. Après l'étape de l'ajout, le prisonnier possède un compte sur l'application mobile 'SKIA pour le placement sous surveillance mobile', il peut créer une session (Authentifier) une et une seule fois (pas de logout) pour commencer le pointage.

4.3.5 Interface 'New User'

Cette interface permet d'ajouter un nouvel agent de police à une région.

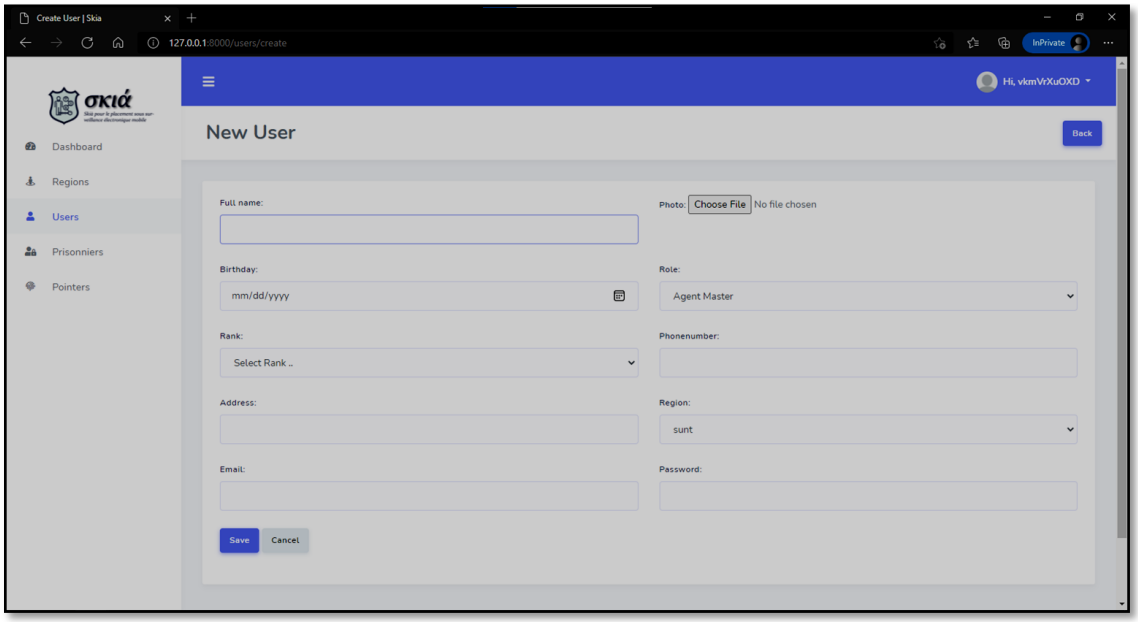


FIGURE 4-7- Interface 'New User'

4.3.6 Interface 'Users'

Cette interface affiche la liste des agents de police avec leurs informations. Elle contient trois boutons:

- *Afficher* : pour le détail de l'agent.
- *Mettre à jour* : pour mettre à jour les informations personnelles de l'agent.
- *Supprimer* : pour supprimer l'agent.

Au coin supérieur droit, on trouve un bouton qui permet d'ajouter un nouvel agent.

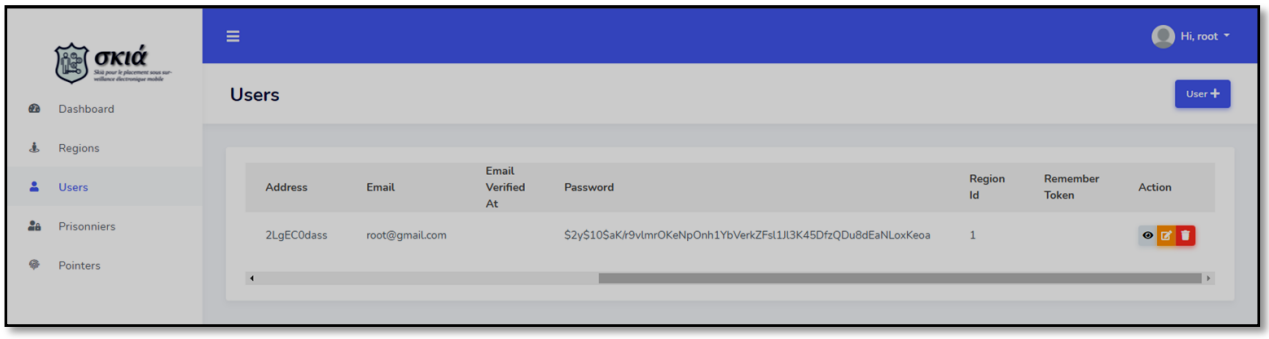


FIGURE 4-8- Interface 'Users'

4.3.7 Interface 'Régions'

Cette interface permet de gérer les régions (mettre à jour et supprimer).

Au coin supérieur droit, on trouve un bouton qui permet d'ajouter une nouvelle région.

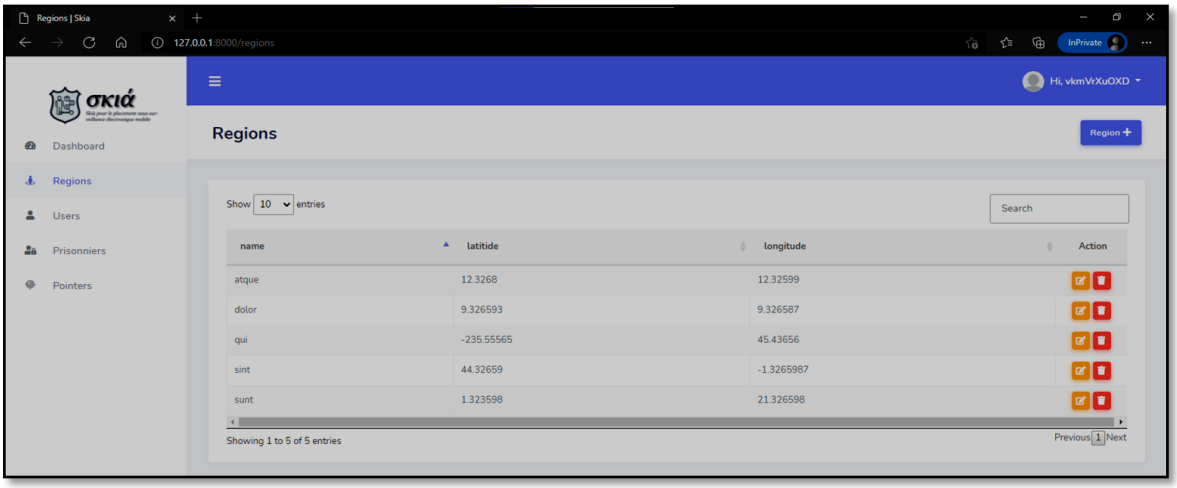


FIGURE 4-9- Interface 'Regions'

4.3.8 Interface 'Prisonniers'

Cette interface permet de gérer les prisonniers (Afficher, mettre à jours et supprimer).

Au coin supérieur droit, on trouve un bouton qui permet d'ajouter un nouveau prisonnier.

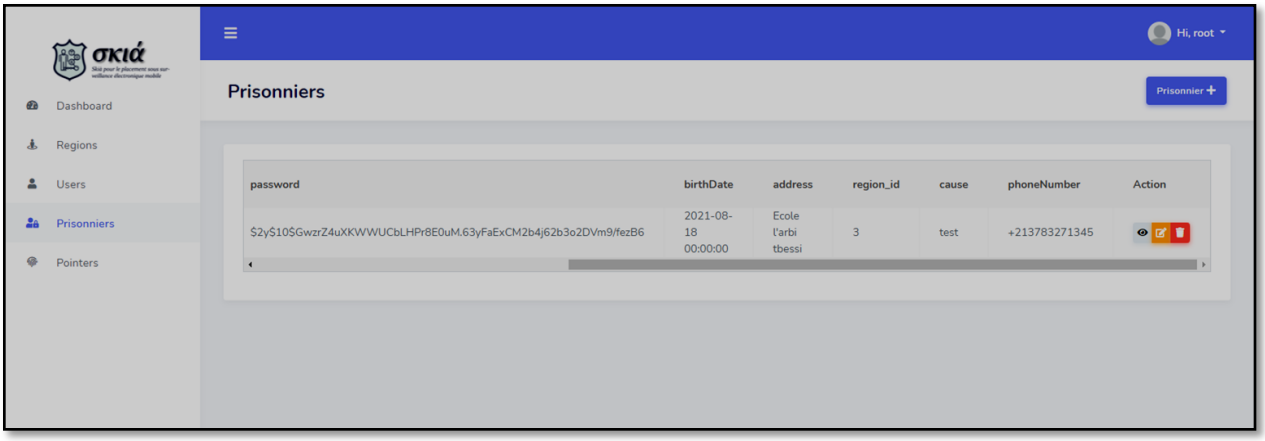


FIGURE 4-10- Interface 'Prisonniers'

4.3.9 Interface 'Login ' Application mobile –Agent de police-

Si l'agent de police possède déjà un compte, il saisit juste correctement son Email et son Password, ensuite il clique sur '**Login**'.

Cette interface 'login' permet à l'agent de police de créer une session- d'authentifier- afin d'accéder à l'interface accueil.

FIGURE 4.11- Interface 'Login' Application mobile –Agent de police-

Les codes ci-dessous expliquent le fonctionnement du Login :

a. Code dans le coté mobile (flutter) :

```
29  Future<void> login() async {
30    Map jsonMapEmail = {
31      "username": _username_controller.text,
32      "password": _password_controller.text,
33    };
34    String replyApi = await apiRequest(
35        "http://192.168.1.6:5000/api/v0.1/agent/login", jsonMapEmail);
36
37    controller_auth.token_value(replyApi);
38    Get.off(Home(), arguments: replyApi);
39  }
```

FIGURE 4.12- Code (*flutter*) 'créer une session Agent de police –login-'

a. *Code dans le coté serveur (flask) :*

```python
@app.route('/api/v0.1/agent/login', methods=['POST'])
def loginAgent():
    body = request.get_json(force = True)

    _username = body.get('username')
    _password = body.get('password')

    if not _username or not _password:
        return make_response('Could not verify', 401, {'WWW-Authenticate': 'Basic realm="Login required! "'})

    user = db.session.query(Agent).filter_by(fullName=_username).first()

    if not user:
        return make_response('Could not verify ', 401, {'WWW-Authenticate': 'Basic realm="Login required! "'})

    if bcrypt.check_password_hash(user.password.encode('utf8'), body.get('password').encode('utf8')):
        token = jwt.encode({'id': user.id, 'exp' : datetime.datetime.utcnow() + datetime.timedelta(minutes=180)}, app.config['SECRET_KEY'])
        # print(token)

        return jsonify({
            'token': token.decode('UTF-8'),
            "message": "login success !!"
        })

    return make_response('Could not verify ', 401, {'WWW-Authenticate': 'Basic realm="Login required!"'})

```

FIGURE 4.13- Code (*flask*) 'créer une session Agent de police –login-'

4.3.10 Interface 'Accueil ' Application mobile –Agent de police-

Cette interface contient un Map qui montre les coordonnées exactes du prisonnier à l'agent de police dans le cas ou le prisonnier dépasse la région verte.

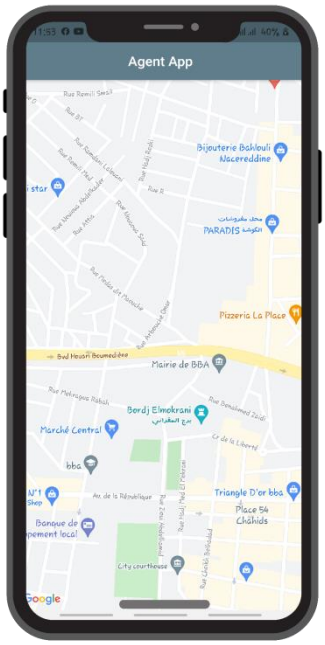


FIGURE 4.14- Interface 'accueil' Application mobile –Agent de police-

4.3.11 Interface 'Login' Application mobile –Prisonnier-

Si le prisonnier possède déjà un compte, il saisit juste correctement son Email et son Password, ensuite il clique sur '**Login**'.

Cette interface 'login' permet au prisonnier de créer une session- d'authentifier- afin de commencer le pointage et de recevoir les notifications.

Cette étape –la création d'une session (l'authentification) se fait devant l'agent master et aussi une et une seul fois (pas logout)).

FIGURE 4.15- Interface 'Login' Application mobile –Prisonnier-

4.3.12 Interface 'Signaler présence ' Application mobile -Prisonnier-

Après le reçu de la notification du pointage, le prisonnier pose son doigt sur son smartphone pour la lecture de son empreinte. L'application envoie l'emplacement du prisonnier au serveur du traitement.

FIGURE 4.16- Interface 'Signaler présence' Application mobile –Prisonnier-

Les codes ci-dessous expliquent le cas 'signaler présence' :

a. *Code dans le coté mobile (flutter) :*

```
14  static Future<List<Prisonnier>?> postPointer() async {
15    var token = controller_auth.Token;
16    int user_id = controller_auth.Id;
17
18    DateTime now = DateTime.now();
19    String formattedDate = DateFormat('yyyy-MM-dd hh:mm:ss').format(now);
20    var userLocation = await location.getLocation();
21    var response = await client.post(
22        Uri.parse('http://192.168.1.4:5000/api/v0.1/prisonnier/pointer'),
23        headers: {"x-access-token": token},
24        body: jsonEncode(<String, String>{
25          "fingrID": "1",
26          "pointAt": formattedDate,
27          "latitude": userLocation.latitude.toString(),
28          "longitude": userLocation.longitude.toString(),
29          "prionnier_id": user_id.toString()
30        }));
31    if (response.statusCode == 200) {
32      var response_json = json.decode(response.body);
33      var data = json.encode(response_json["data"]);
34      return prisonnierFromJson(data);
35    } else {
36      return null;
37    }
38  }
```

FIGURE 4.17- Code (*flutter*) 'signaler présence'

b. Code dans le coté serveur (flask) :

```python
@app.route('/api/v0.1/prisonnier/pointer', methods=['POST'])
@token_required_prisonnier
def pointer(current_user):
    body = request.get_json(force = True)

    _faceID = body.get('faceID')
    _fingrID = body.get('fingrID')
    _pointAt = body.get('pointAt')
    _latitude = body.get('latitude')
    _longitude = body.get('longitude')
    _prionnier_id = body.get('prionnier_id')
    if(_faceID)and(_fingrID):
        newPointer = Pointer(faceID=_faceID, fingrID=_fingrID,pointAt=_pointAt,latitude=_latitude,longitude=_longitude,prionnier_id=_prionnier_id)

        db.session.add(newPointer)
        db.session.commit()

        return jsonify({
        "message": "Pointer success !!"
            })
    else:
        return jsonify({
        "message": "Pointer Error !!"
            })

```

FIGURE 4.18- Code (*flask*) 'signaler présence'

c. Pour afficher la notification

```dart
    int ID, String title, String body) async {
  await AwesomeNotifications().createNotification(
    content: NotificationContent(
      id: ID,
      channelKey: 'basic_channel',
      title: '${Emojis.time_alarm_clock} $title !!!',
      body: body.toString(),
      notificationLayout: NotificationLayout.Default,
    ),
  );
 }
}

```

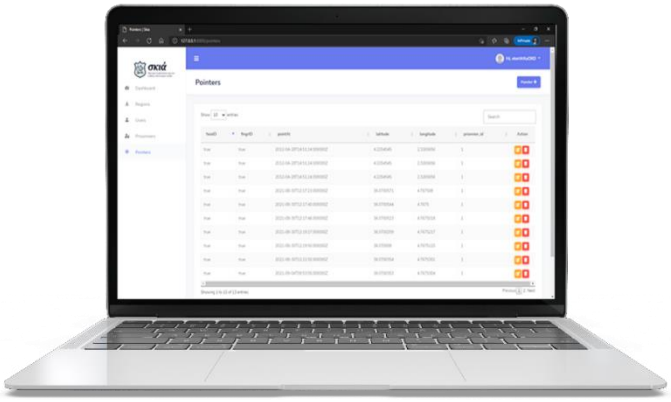
FIGURE 4.19- Code (*flutter*) 'afficher notification'

Le serveur traite et compare l'empreinte du pointage avec celle déjà enregistrée dans la base de données (le traitement et la comparaison se font par un système de reconnaissance d'empreinte digitale (chapitre1)) et calcule l'emplacement du prisonnier.

1- Une vérification réussie :

Le serveur du traitement envoie le résultat de la vérification ‘réussie’ à l’agent master et émet un ‘event’ de l’emplacement du prisonnier au serveur websocket, ce dernier envoie une notification de l’emplacement du prisonnier dans la région verte à l’agent master. Le prisonnier sait que l'opération est réussie quand l’interface du pointage disparaît.

L’agent master peut consulter la liste des prisonniers présents et gérer ces derniers (mettre à jour et supprimer).



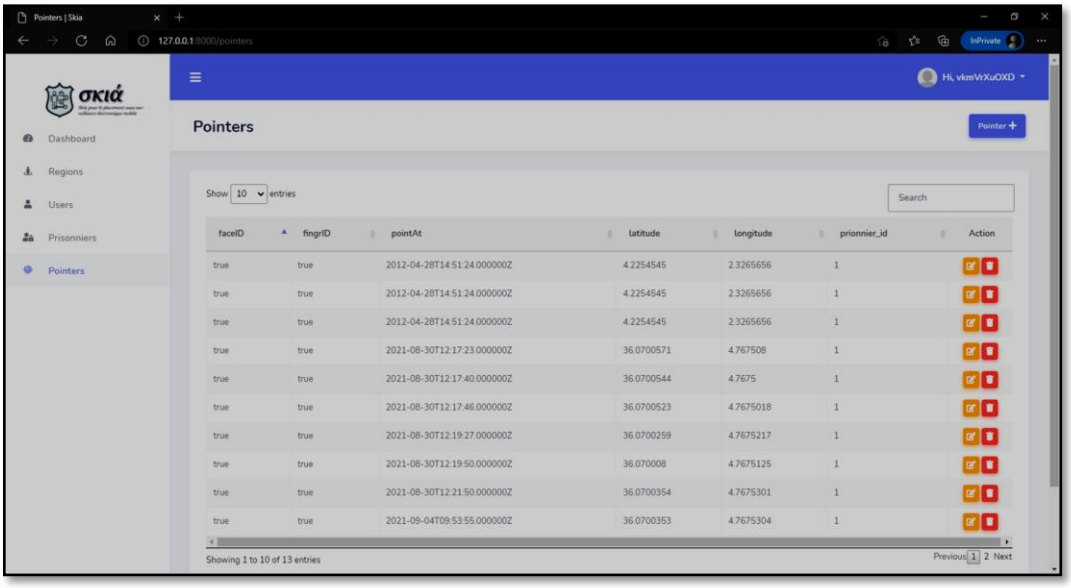


FIGURE 4.20- Gestion du pointage

2- Une mauvaise présence :

Le serveur envoie le résultat de la vérification échouée à l'agent master et à l'agent de police et émet un 'event' de l'emplacement du prisonnier au serveur websocket, ce dernier envoie une notification de l'emplacement du prisonnier qui dépasse la région verte à l'agent master et à l'agent de police, et au prisonnier.

Le prisonnier peut pointer d'une façon incorrecte, il dépasse le temps du pointage ou bien il clique sur le bouton cancel de l'authentification, dans ce cas l'agent master et l'agent de police reçoivent une notification 'erreur du pointage'.

Dans les trois cas 'verification échouée, erreur du pointage et le condamné dépasse la région verte) l'agent master et l'agent de police prennent les mesures nécessaires.

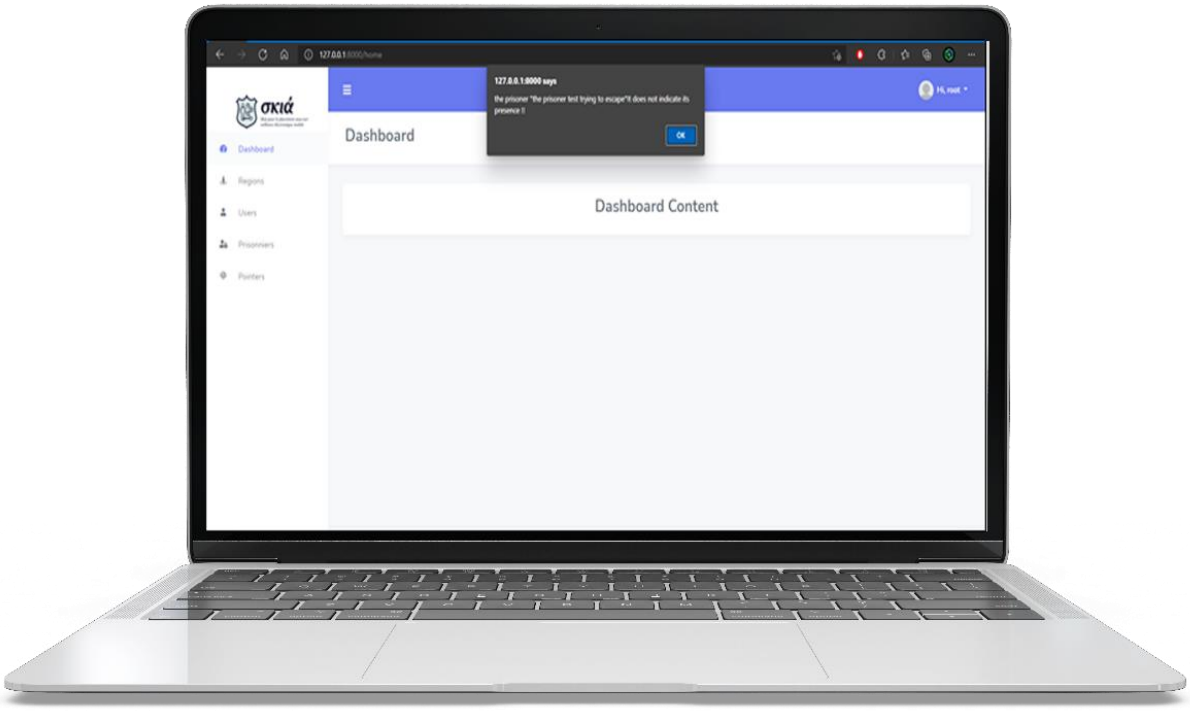


FIGURE 4.21- Notification 'erreur du pointage'

Les codes ci-dessous expliquent l'implémentation du 'Notification' :

a. Code au niveau du serveur :

```
24
25      client.on('escape', function(data) {
26          console.log("",data);
27          client.broadcast.emit('escape_msg', 'the prisoner '+data+'trying to escape!!');
28          client.broadcast.emit('pointage_errore', 'the prisoner "'+data+'"It does not indicate its presence
29       });
30
```

FIGURE 4.22- Code (NodeJS) pour afficher la notification

b. Code au niveau du site web :

```
51  @include('profile.edit_profile')
52  <script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/4.2.0/socket.io.js"></script>
53   <script>
54          var socket = io.connect('http://192.168.1.6:4200');
55          socket.on('connect', function (data) {
56              socket.emit('pointage_error', 'Hello World from Agent web site',);
57          });
58          socket.on('pointage_errore', function (data) {
59              console.log(data);
60              alert(data);
61          });
62
63
```

FIGURE 4.23- Code (Laravel) pour afficher la notification

L'application mobile 'prisonnier' permet de déterminer l'emplacement du prisonnier, à chaque instant (en temps réel), les régions sont divisé en 3 régions : verte, orange et rouge.

FIGURE 4.24- Notification de l'emplacement (région orange)

FIGURE 4.25- Notification de l'emplacement (région rouge)

C'est le serveur qui divise les régions en trois régions, et qui calcule l'emplacement exact du prisonnier par rapport aux trois régions, le code ci-dessous explique ça:

```
@sio.on('position')
def message(data):
    distence =calculeDistance(float(data['latitude_user']), float(data['longitude_user']), float(data['latitude']), float(data['longitude']))
    # if ((distence<=500.00)):
        # sio.emit('distence', "you are in Green Region")
    if (distence>100.00)and(distence<=150.00):
        sio.emit('distence', "you are in Orange Region")
    elif (distence>150.00):
        time.sleep(2)
        sio.emit('distence', "you are in Red Region")
```

FIGURE 4.26- Code (flask) dedevise la région

4.4 Conclusion

La phase de réalisation est l'étape la plus importante dans le cycle de vie d'une application. Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre application en spécifiant l'environnement, les outils et les langages de développement associés à notre système. En effet, nous avons achevé l'implémentation tout en respectant la conception élaborée.

Conclusion générale

& perspectives

Conclusion générale & perspectives

Avec le développement accéléré de l'informatique, les applications mobiles deviennent de plus en plus utilisées dans pratiquement tous les secteurs, notamment la surveillance électronique.

Notre projet s'inscrit justement dans ce cadre, il s'agit de remplacer le bracelet électronique par une application mobile sous Android et iOS, comme un moyen de surveillance des condamnés, cette application mobile s'avère être un alternatif plus fiable et plus efficace, qui utilise des données biométriques propres à chaque individu (l'empreinte digitale). Cette nouvelle solution nous a permis de supprimer l'effet psychologique présenté par le bracelet, et de mieux protéger la vie privée des prisonniers. Certainement, le nouveau système n'est pas parfait mais il présente beaucoup d'avantages par rapport au premier, et peut donc offrir une expérience plus flexible.

Cette application nommée « Skia pour le placement sous surveillance mobile», qui a pour rôle la collecte et la transmission régulière des empreintes digitales des prisonniers à un centre de contrôle, dans le but de s'assurer de la présence de ces derniers à chaque instant et en tous lieux. Le travail porte sur l'amélioration des applications mobiles de surveillance des prisonniers sous Android et iOS qui consistent à contrôler via un smartphone si effectivement un condamné est chez lui ou pas. L'application doit être hébergée dans un serveur distant. Un smartphone est délivré au condamné avec l'application client installée. Via la connexion internet, on demande au condamné de se connecter au serveur pour enregistrer ses empreintes digitales. De façon régulière, le condamné doit activer sa présence en posant juste le doigt sur son smartphone pour la lecture et la vérification d'empreintes.

Pour cela, nous avons en premier lieu présenté les deux domaines : la biométrie et la surveillance électronique (SE), une description du cadre du projet et la méthodologie de conception en l'occurrence UML comme langage de modélisation . Nous avons établi par la suite, une étude préliminaire pour identifier les différents acteurs qui interagissent avec le système à réaliser, suivi de la spécification des besoins fonctionnels à travers un diagramme de cas d'utilisation, de séquence et de classe.

Enfin, les outils et les langages de développement mobile que nous avons utilisé pour implémenter notre application ont été exposés.

Ce projet nous a été très bénéfique, car nous avons enrichi nos connaissances sur les deux plans : théorique et pratique. Il nous a aussi permis de découvrir et d'acquérir de nouvelles connaissances en matière de développement mobile.

Finalement on peut imaginer de nombreuses perspectives pour améliorer ce système, on peut citer par exemple :

1. L'ajout des autres modalités plus complexes peuvent être rajoutées,
2. plusieurs scénarios possibles doivent être développés,

Tout ça va augmenter la fiabilité, et si on arrive à ce stade on peut envisager l'utilisation de ce nouveau moyen pour une excellente surveillance des prisonniers à tout moment en tous lieux avec confiance.

Références

Références

[1] J.Mahier, M. Pasquet, C. Rosenberger, and F. Cuozzo, « Biométric authentification », Encyclopedia of Iinformation Science and Technology, 2008

[2] J.Wayman,"Biométric Systems Technology Design and Perfermance Evaluation",London, 2005

[3] A.K. Jain, S. Prabhakar and S. Pankanti, "Twin Test: On Discriminability of Fingerprints", Proc. 3rd International Conference on Audio- and Video-Based Person Authentication,, pp. 211-216, Sweden, June 6-8, 2001

[4] H. Ailisto and M. Linholm, "A reviex of fingerprint image enhancement methods", International Journal of Image and Graphics, Vol. 3, No. 3, pp. 401-424, 2003

[5] A.Ross, A.K.Jain, and Engineering, "Information fusion In biometrics", Departement of computer Science, Michigan State University, Pattern Recognition, Letters, 2003

[6] Etude d'un système complet de reconnaissance d'empreintes digitales pour un capteur microsystème à balayage . GALY, Nicolas. 2005

[7] Fingerprint verification based on minutiae features: a review", Pattern Analysis and Applications, Vol. 7, No. 1, pp. 94-113. Amin, N. Yager and A. 2004

[8] https://www.cairn.info/revue-mouvements-2014-3-page-109.htm

[9] https://www.cairn.info/la-surveillance-electronique--9789287180971-page-17.htm

[10] https://www.institutpourlajustice.org/content/2017/11/Etudes-Criminologie- le_ bracelet _ electronique _ mobile_previent-il_efficacement_la_recidive.pdf

[11]https://usbeketrica.com/fr/article/surveillance-electronique-nouveaux-bracelets-en-debat

[12] https://www.lejdc.fr/nevers-58000/actualites/comment-fonctionne-le-placement-sous-bracelet - electronique-on-vous-explique_13664213/

[13] Chantal morley, Jean hugues, Bernard le blanc. UML2, pour l'analyse d'un Système d'information 4e édition, 2009

[14] http://projet.eu.org/pedago/sin/term/5-architecture_MVC.pdf

[15] The JWT Handbook,Sebastián E. Peyrott, Autho Inc. Version 0.14.1

[16] https://jwt.io/introduction

[17] Pascal Roques, Les cahiers du programmeur UML2 modélisé une application web, Eyrolles, 2007,4ème édition

[18] https://lms.fun-mooc.fr/c4x/usousse/74001S02/asset/Lecon_4.1.pdf

[19] https://code.visualstudio.com/docs

[20]https://www.microsoft.com/enus/p/windowsterminal/9n0dx20hk701?activetab=pivot:overviewtab

[21] https://www.adobe.com/fr/products/illustrator.html

[22] https://www.postman.com/

[23] https://www.apachefriends.org/fr/index.html

[24] https://dart.dev/overview

[25] https://docs.python.org/fr/3/tutorial/

[26]https://developer.mozilla.org/fr/docs/Learn/JavaScript/First_steps/What_is_JavaScript

[27] https://www.php.net/

[28]https://medium.com/@concisesoftware/what-is-flutter-here-is-everything-you-should-know-faed3836253f

[29] https://pythonbasics.org/what-is-flask-python/

[30] https://www.codecademy.com/learn/learn-node-js

[31]https://careerfoundry.com/en/blog/web-development/what-is-bootstrap-a-beginners-guide/

[32] https://laravel.com/docs/8.x

Résumé

L'importance que sont en train de prendre les applications mobiles de surveillance attire l'attention. Ce type d'applications mobiles contribue à améliorer la qualité de vie des personnes, grâce aux possibilités de surveillance et au suivi qu'elles offrent. Ce projet consiste à remplacer le bracelet électronique ou le placement sous surveillance électronique (PSE) (une mesure d'aménagement de peine permettant d'exécuter une peine d'emprisonnement sans être incarcéré) par une application sous Android et iOS qui puisse faire la même chose. Cette application nommée "skia pour le placement sous surveillance mobile", permet de contrôler via un smartphone si effectivement un condamné est chez lui ou pas. L'application doit être hébergée dans un serveur distant. Un smartphone est délivré au condamné avec l'application client installée. Via la connexion internet, on demande au condamné de se connecter au serveur pour enregistrer ses empreintes digitales. De façon régulière, le condamné doit activer sa présence en posant juste le doigt sur son smartphone pour la lecture et la vérification d'empreintes. Afin de réaliser notre application, nous avons utilisé plusieurs outils et lagunages de programmation, citons :UML, Flutter, Dart, Postman, python, XAMPP, Flask, laravel, Bootstrap, Node js….

Mots clés : Condamné, bracelet électronique, application mobile, empreinte digitale, UML, Android, Flutter, Dart, Postman, python, XAMPP, Flask, laravel, Bootstrap, Node js.

Abstract

The growing importance of mobile surveillance applications is drawing attention. This type of mobile application helps improve people's quality of life, thanks to the monitoring and tracking possibilities they offer. This project is to replace the electronic bracelet or placement under electronic surveillance (PSE) (a sentence adjustment measure to serve a prison sentence without being incarcerated) by an application running Android and iOS that can do the same. This application called "skia for placement under mobile surveillance", allows you to control via a smartphone whether indeed a convict is at home or not. The application must be hosted on a remote server. A smartphone is issued to the convict with the client application installed. Via the internet connection, the convict is asked to connect to the server to register his fingerprints. On a regular basis, the convict must activate his presence by just placing his finger on his smartphone for the reading and verification of fingerprints. In order to realize our application, we used several tools and programming language, like: UML, Flutter, Dart, Postman, python, XAMPP, Flask, laravel, Bootstrap, Node js.…

Keywords: Condemned, electronic bracelet, mobile application, fingerprint, UML, Android, Flutter, Dart, Postman, python, XAMPP, Flask, laravel, Bootstrap, Node js.

الملخص

تجذب الأهمية المتزايدة لتطبيقات المراقبة المتنقلة الانتباه. يساعد هذا النوع من تطبيقات الهاتف المحمول على تحسين نوعية حياة الناس ، وذلك بفضل إمكانيات المراقبة والتتبع التي يقدمونها. يهدف هذا المشروع إلى استبدال السوار الإلكتروني أو الوضع تحت المراقبة الإلكترونية (PSE) -كإجراء لتعديل عقوبة السجن دون أن يتم سجنه- بواسطة تطبيق يعمل بنظام Android و iOS والذي يمكنه فعل الشيء نفسه. يتيح لك هذا التطبيق المسمى 'skia pour le placement sous surveillance mobile' عبر الهاتف الذكي التحقق في ما إذا كان المدان في المنزل بالفعل أم لا. يتم إعطاء هاتف ذكي للمحكوم عليه مع تثبيت التطبيق. بواسطة الإنترنت ، يُطلب من المحكوم عليه الاتصال بالserveur لتسجيل بصمات أصابعه. على أساس منتظم ، يجب على المحكوم عليه إثبات وجوده وهذا بمجرد وضع إصبعه على هاتفه الذكي لقراءة بصمات الأصابع والتحقق منها. من أجل تحقيق تطبيقنا ، استخدمنا العديد من الأدوات ولغات البرمجة نذكر منها: UML،Flutter،Dart،Postman ،python ،XAMPP ،Flask ،Laravel ،Bootstrap ،Node js.

الكلمات المفتاحية: مدان ،سوار إلكتروني ،تطبيق جوال ،بصمة ، UML، Android، Flutter، Dart، Postman، python، XAMPP، Flask، laravel، Bootstrap، Node js.