

## Table des matières

I.	Introduction.....	2
a.	Présentation du projet .....	2
b.	Objectifs.....	3
c.	Méthodologie utilisée .....	4
d.	Gestion du projet avec une structure de dossiers organisée.....	5
e.	Gestion du projet avec un diagramme de Gantt.....	5
•	.....	6
II.	Architecture et conception du système .....	7
f.	Description de la solution proposée .....	7
g.	Diagramme des cas d'utilisation .....	8
h.	Architecture globale du système.....	9
i.	Rôle du transistor IRF520 .....	11
III.	Conception de la base de données .....	12
j.	Le rôle de la base de données synchronisée .....	12
k.	Hébergement de la base de données.....	12
l.	Processus de conception .....	13
m.	Evènements phpmyadmin.....	16
IV.	Conception de l'interface de gestion (IG).....	18
n.	Fonctionnalités de l'interface de gestion .....	18
o.	Demandes d'accès .....	20
p.	Envoi des courriels électroniques.....	23
q.	Génération des QR-Codes .....	25
r.	Visualisation de l'état d'occupation des ressources.....	28
s.	Historique d'accès aux ressources .....	30
t.	Gestion de l'accès par reconnaissance faciale .....	31
u.	Accès aux serrures .....	33
v.	Conception visuelle : L'optimisation de l'interface grâce au CSS.....	34

## **I. Introduction**

### **a. Présentation du projet**

Le contrôle d'accès est un enjeu majeur pour la sécurité des bâtiments, notamment dans les institutions judiciaires. C'est dans ce contexte que s'inscrit le présent projet, qui vise à développer une solution de contrôle d'accès pour le tribunal de grande instance d'Evry.

Actuellement, l'accès est géré par un vigile qui délivre des badges permettant de se déplacer librement dans le bâtiment. Cependant, ce système présente une faille de sécurité majeure, permettant un accès non autorisé à des zones sensibles. Afin de résoudre ce problème, le système Contrôle-Accès a été conçu pour renforcer la sécurité et simplifier la gestion des accès.

L'objectif principal de cette solution est de donner aux responsables du tribunal un contrôle total sur les accès en décidant qui peut accéder où et quand. Grâce à l'utilisation de QR codes d'accès temporaires ou permanents, les utilisateurs peuvent accéder aux ressources du tribunal uniquement pendant les périodes autorisées. De plus, l'introduction d'un contrôle biométrique pour l'accès aux locaux sensibles garantit une sécurité renforcée. Cette mesure de sécurité supplémentaire consiste en une reconnaissance faciale pour les dossiers et les pièces à conviction des affaires criminelles.

En somme, le projet Contrôle-Accès pour le tribunal de grande instance d'Evry constitue une avancée majeure dans la gestion de la sécurité et des accès. En renforçant la sécurité, en simplifiant la gestion des accès et en offrant une visibilité complète sur les flux de personnes, cette solution contribue à garantir l'intégrité des ressources du tribunal et la protection des utilisateurs et des biens.

Le présent document expose les spécifications de la solution de contrôle d'accès pour le tribunal de grande instance d'Evry, y compris les diagrammes SysML et les contraintes de réalisation. Le projet ne prend pas en compte la gestion des dysfonctionnements possibles sur le système embarqué, tels que la panne de la serrure ou de la caméra.

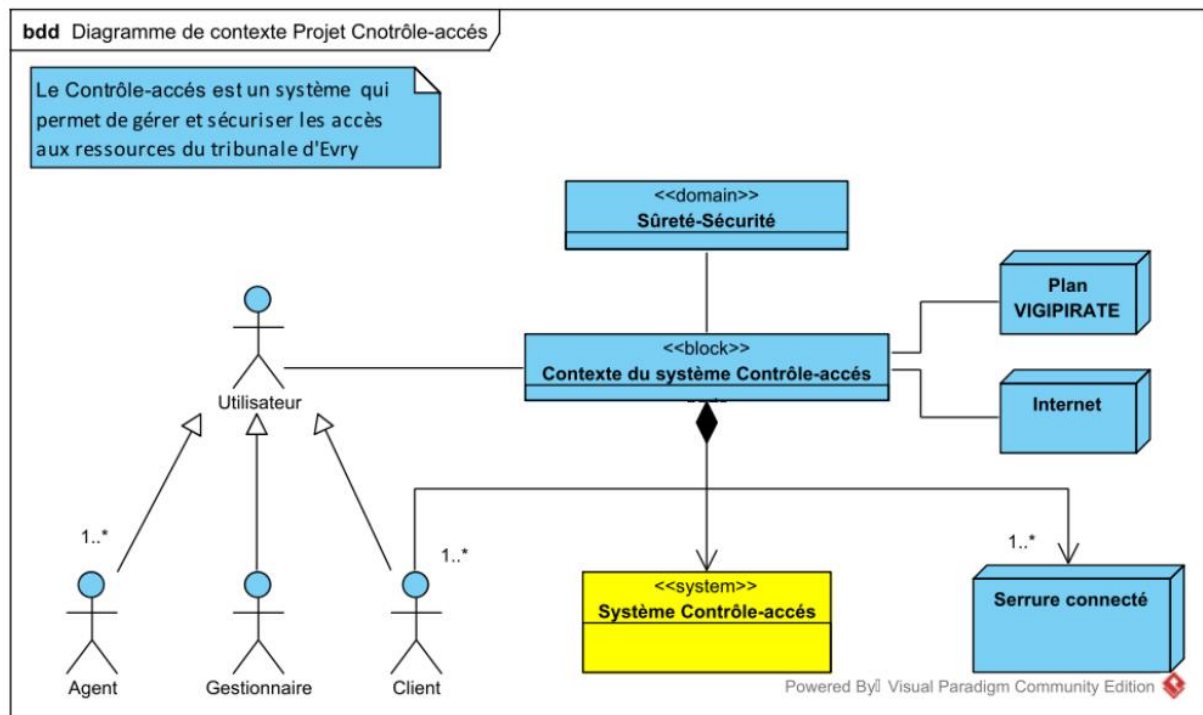


Figure 1. Diagramme de contexte

## b. Objectifs

Les objectifs du projet sont les suivants :

- Développer une solution de contrôle d'accès pour le tribunal de grande instance d'Evry.
- Permettre la gestion à distance de l'accès aux ressources du tribunal, notamment les immeubles, les étages, les locaux et les salles d'audience.
- Simplifier la gestion et le contrôle des accès en permettant de décider qui peut accéder, où et quand.
- Fournir des QR-codes d'accès temporaires ou permanents aux utilisateurs pour accéder aux ressources uniquement pendant les périodes autorisées.
- Exiger un contrôle biométrique pour l'accès aux locaux sensibles, tels que ceux réservés pour les dossiers et les pièces à conviction des affaires criminelles.
- Remplacer les clés et les badges par des QR-codes pour réduire les pertes de supports et limiter les usages abusifs.
- Limiter les accès temporaires ou dérogatoires aux périodes autorisées.
- Offrir une vue globale et synthétique sur le niveau d'occupation des ressources, ainsi que l'historique des accès en temps réel.
- Développer trois sous-systèmes : l'interface de gestion, la serrure QR-Code et la serrure biométrique par reconnaissance faciale.

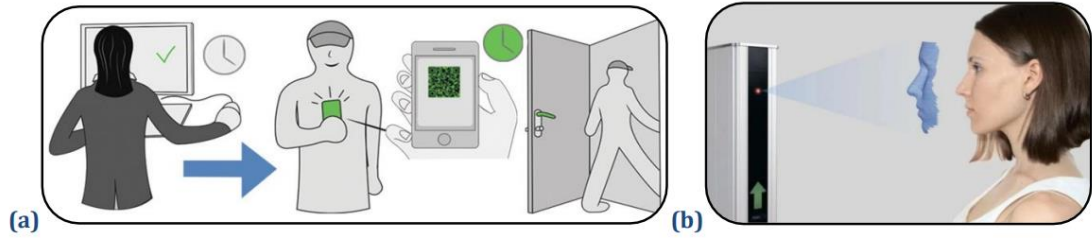


Figure 2. (a) Accès avec QR-code.

(b) Contrôle d'accès par reconnaissance faciale.

### c. Méthodologie utilisée

Pour répondre aux objectifs du projet, nous avons utilisé une méthodologie en plusieurs étapes :

- **Etape 1. Analyse des besoins** : L'analyse des besoins consiste à identifier les besoins du client et les spécifications fonctionnelles et techniques du système de contrôle d'accès. Cette étape comprend la rédaction du cahier des charges et la réalisation des diagrammes SysML, tels que le diagramme de contexte, le diagramme des cas d'utilisation, le diagramme d'exigence et le diagramme de séquence. Elle permet de comprendre les besoins des utilisateurs et les fonctionnalités du système de contrôle d'accès.
- **Etape 2. Conception** : La conception consiste à concevoir l'architecture globale du système, ainsi que les sous-systèmes de l'interface de gestion, de la serrure QR-Code et de la serrure biométrique par reconnaissance faciale. Cette étape permet de déterminer les différentes fonctionnalités de chaque sous-système et de s'assurer de leur intégration harmonieuse.
- **Etape 3. Réalisation** : La réalisation consiste à mettre en place la plateforme de développement et à programmer les différents sous-systèmes. Cette étape inclut également les tests de validation de chaque composant et du système global. Elle permet de vérifier la conformité du système aux spécifications fonctionnelles et techniques définies lors de l'analyse des besoins.
- **Etape 4. Tests** : Une fois les sous-systèmes développés et intégrés, un test de validation sera effectué pour vérifier le bon fonctionnement de l'ensemble du système. Le test permettra de s'assurer que les serrures s'ouvrent et se ferment correctement, que les QR-codes et la reconnaissance faciale fonctionnent correctement, que les autorisations d'accès sont bien prises en compte et que l'interface de gestion offre une vue synthétique et en temps réel sur l'état du système. Des scénarios de test seront élaborés pour couvrir différents cas d'utilisation possibles.

#### d. Gestion du projet avec une structure de dossiers organisée

Pour gérer le projet, j'ai créé un dossier sur mon PC que j'ai synchronisé avec Google Drive afin que mes camarades puissent également y accéder. Ce dossier est organisé en quatre sous-dossiers principaux : "01 - Conception et planification", "02 - Développement", "03 - Test" et "04 - Documentation finale".

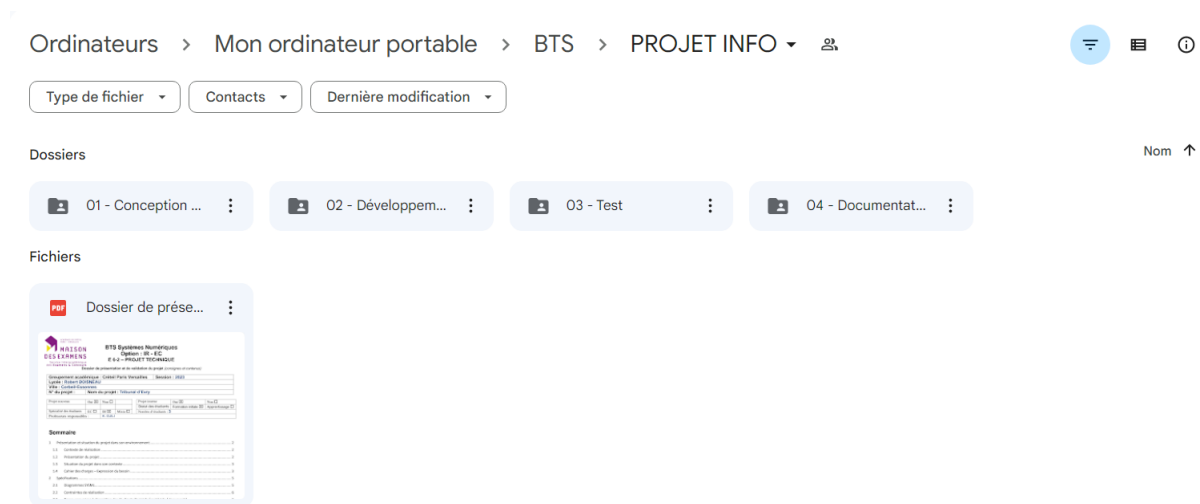


Figure 3. Organisation des dossiers du projet

Nous avons organisé le projet en utilisant quatre sous-dossiers principaux. Le premier, "Conception et planification", contenait tous les documents liés à la phase initiale du projet, tels que les spécifications et les diagrammes. Le deuxième, "Développement", regroupait tous les fichiers liés à l'écriture du code et à la création des fichiers HTML, CSS et JavaScript. Le troisième, "Test", était dédié aux tests pour vérifier le bon fonctionnement du site web et corriger les éventuels problèmes. Enfin, le quatrième, "Documentation finale", rassemblait tous les fichiers et documents nécessaires pour présenter le projet de manière complète et professionnelle. Cette organisation nous a permis de travailler de manière structurée et de faciliter la collaboration entre les membres de l'équipe.

#### e. Gestion du projet avec un diagramme de Gantt

Nous avons utilisé un diagramme de Gantt pour planifier et organiser les différentes étapes de notre projet. Le diagramme de Gantt nous a permis d'avoir une vision claire et chronologique des tâches à réaliser. Voici quelques exemples de tâches présentes dans le diagramme :

- "Access Tribunal" : Cette tâche était prévue pour durer 17 semaines et 1 jour et était liée à l'accès aux tribunaux.
- "Recherche et documentation" : Cette tâche avait une durée prévue de 2 semaines et 32 minutes et impliquait la recherche et la collecte de documentation pertinente.
- "Conception et planification" : Cette phase du projet était prévue pour durer 4 semaines et 3 jours et comprenait des tâches telles que la conception de l'architecture et l'élaboration des spécifications détaillées.

- "Développement" : Cette phase était prévue pour durer 6 semaines et 2 jours et incluait des tâches spécifiques telles que le développement de l'interface utilisateur, la serrure QR et la serrure biométrique.
- "Test" : Cette phase était prévue pour durer 1 semaine et 3 jours et comprenait des tests unitaires, des tests d'intégration, des tests de performance et des tests utilisateur.
- "Documentation finale" : Cette phase était prévue pour durer 1 semaine et 2 jours et impliquait la rédaction du manuel d'utilisation, de la documentation technique et de la documentation de support.

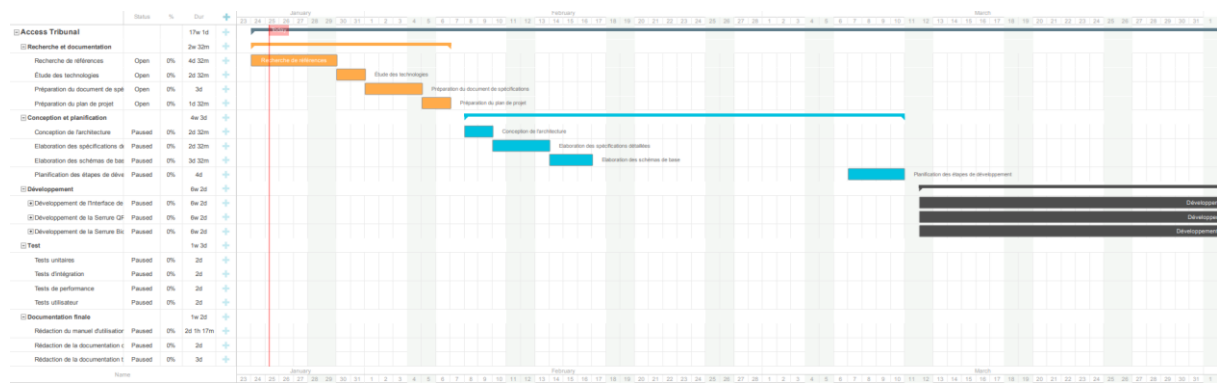
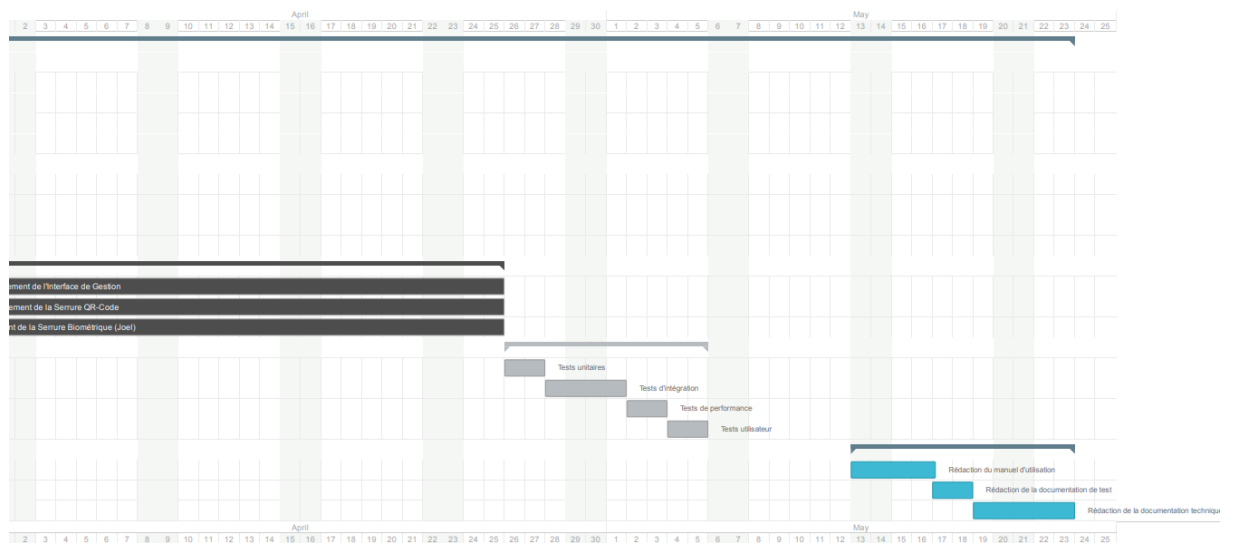


Figure 4. (a) Diagramme de Gantt (1/2)



(b) Diagramme de Gantt (2/2)

Grâce au diagramme de Gantt, nous avons pu visualiser les dépendances entre les différentes tâches, gérer les ressources et respecter les délais fixés pour chaque étape du projet. Cela nous a permis de travailler de manière efficace et de suivre l'avancement du projet de manière claire et organisée.

## II. Architecture et conception du système

### f. Description de la solution proposée

La solution proposée pour le contrôle d'accès au tribunal de grande instance d'Evry comprend trois principaux sous-systèmes interconnectés : l'interface de gestion, la serrure QR-Code et la serrure biométrique par reconnaissance faciale.

- **Etudiant 1 (moi, Omar Hussein), Interface de gestion** : L'interface de gestion est une application web développée en utilisant les langages HTML, CSS, PHP et SQL. Elle permet à l'administrateur de gérer les accès aux différentes ressources du tribunal de manière centralisée. HTML est utilisé pour la structure de la page, CSS pour la mise en forme et le style, PHP pour la logique de gestion des utilisateurs et des accès, et SQL pour interagir avec la base de données.  
L'interface de gestion offre des fonctionnalités telles que la validation des demandes d'accès des utilisateurs, la génération et impression des QR-codes d'accès, la gestion des accès par reconnaissance faciale, la gestion de la messagerie électronique, la visualisation de l'état d'occupation des ressources, la consultation de l'historique des accès et l'accès aux systèmes embarqués des serrures. Elle permet ainsi à l'administrateur de prendre des décisions éclairées sur les autorisations d'accès, de surveiller l'activité en temps réel et de gérer efficacement les utilisateurs et les ressources.
- **Etudiant 2 (Adama Scylla), Serrure QR-Code** : Les serrures QR-Code sont pilotées par des Raspberry Pi à l'aide d'un code Python spécifique. Lorsqu'un utilisateur présente un QR-code valide à la serrure, celle-ci lit et décode le QR-code à l'aide de la caméra intégrée au Raspberry Pi. Si le QR-code est authentifié avec succès, la serrure actionne la gâche de la porte, permettant ainsi l'accès à l'utilisateur. Une LED verte est utilisée pour indiquer l'ouverture de la gâche, tandis qu'une LED rouge indique que la gâche est fermée. Ces indications visuelles permettent de confirmer visuellement si la porte s'est ouverte ou non lors d'une reconnaissance.
- **Etudiant 3 (Joël Nyobe), Serrure biométrique par reconnaissance faciale** : Les serrures biométriques par reconnaissance faciale sont également pilotées par des Raspberry Pi à l'aide d'un code Python spécifique. Lorsqu'un utilisateur se présente devant la serrure, la caméra intégrée au Raspberry Pi capture son visage et utilise un algorithme de reconnaissance faciale pour l'authentifier. Si l'utilisateur est reconnu avec succès, la serrure actionne la gâche de la porte, permettant ainsi l'accès. De même que pour les serrures QR-Code, une LED verte indique l'ouverture de la gâche, tandis qu'une LED rouge indique que la gâche est fermée.

La solution complète de contrôle d'accès offre une gestion centralisée, flexible et sécurisée des accès au tribunal de grande instance d'Evry. Grâce à l'interface de gestion développée en HTML, CSS, PHP et SQL, les administrateurs peuvent prendre des décisions éclairées sur les autorisations d'accès, surveiller l'activité en temps réel et gérer efficacement les utilisateurs et les ressources. Les serrures QR-Code et biométriques, pilotées par des Raspberry Pi à l'aide de code Python, garantissent un contrôle précis des accès et offrent des indications visuelles claires sur l'état de la gâche. Cette solution permet ainsi de renforcer la sécurité du tribunal, de limiter les usages abusifs et de fournir un suivi détaillé de l'historique des accès.

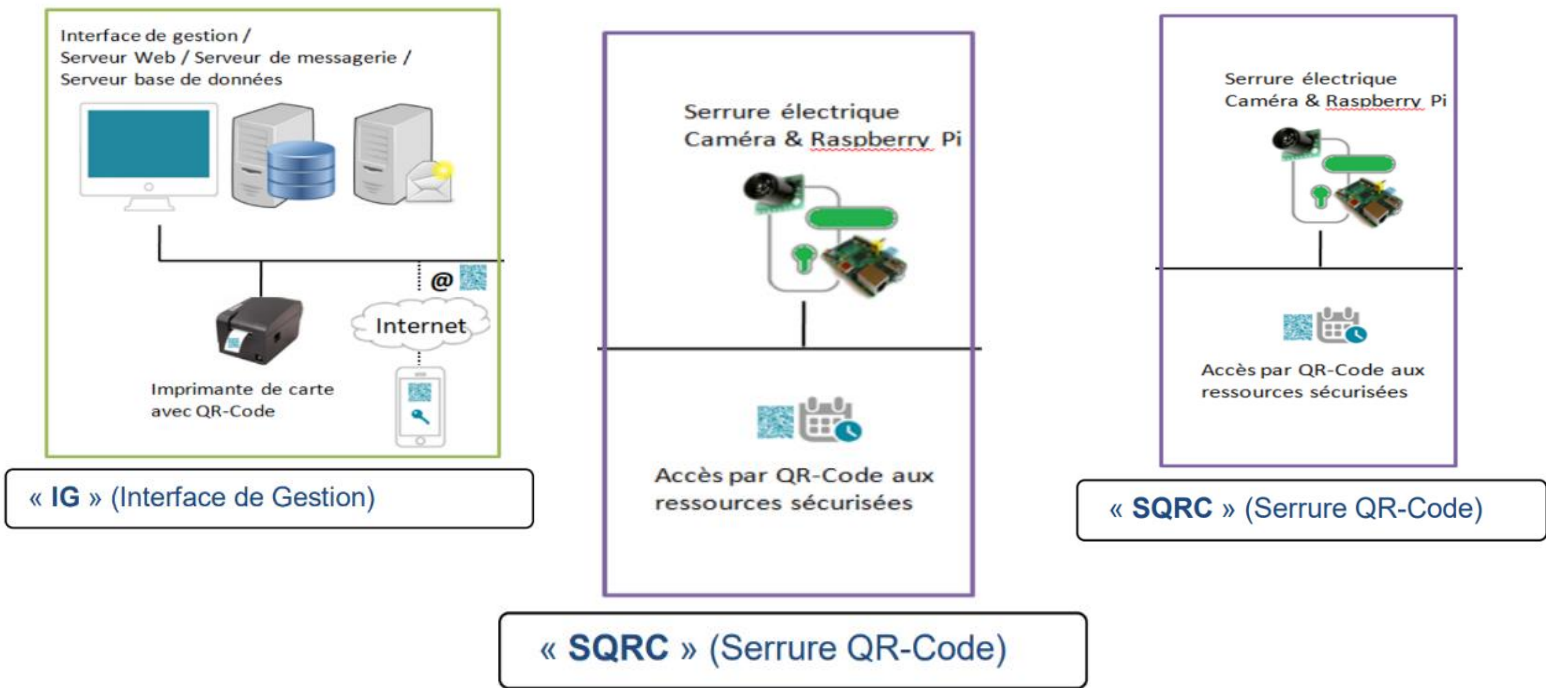


Figure 5. Représentation des trois sous-systèmes.

#### g. Diagramme des cas d'utilisation

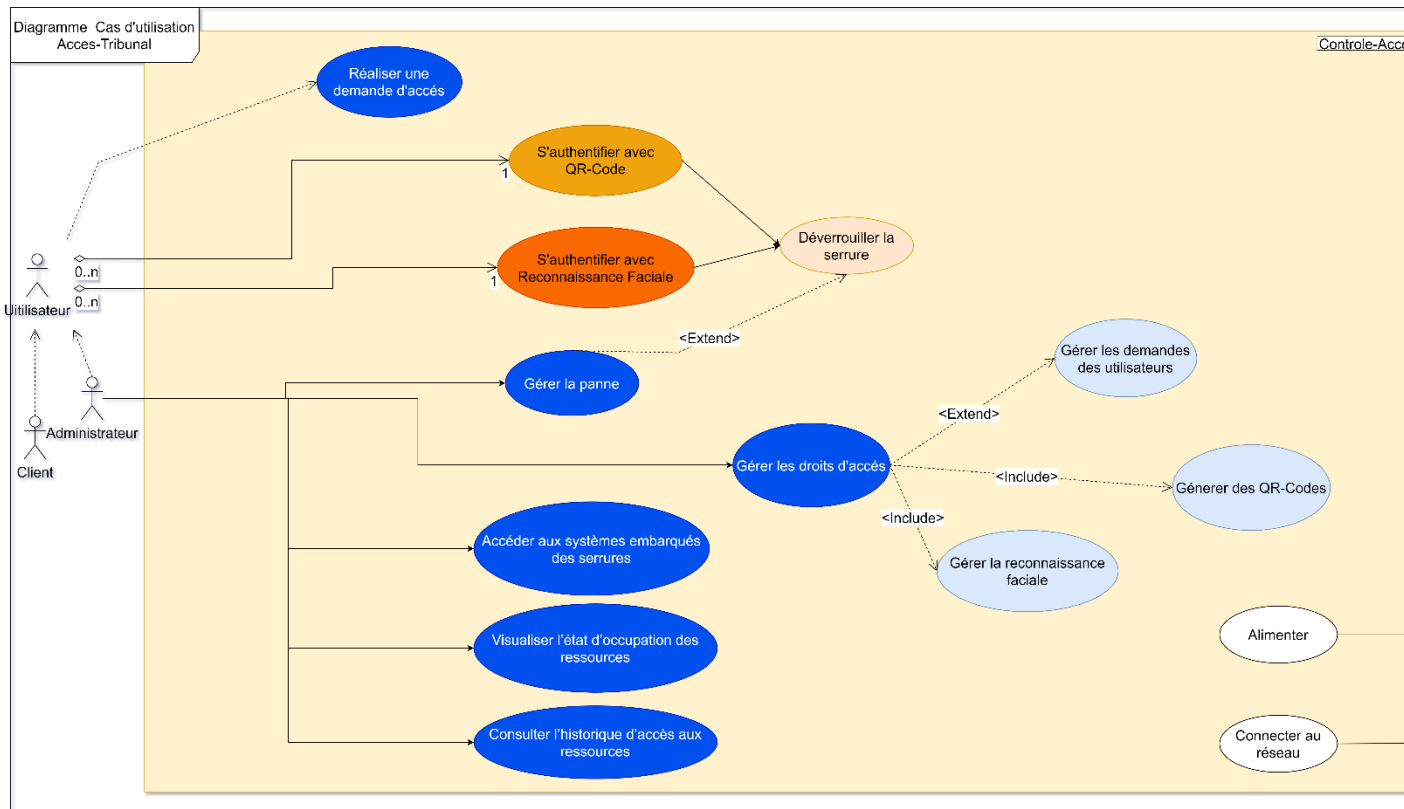


Figure 6. Diagramme des cas d'utilisation

Le diagramme des cas d'utilisation représente de manière visuelle les différentes fonctionnalités et interactions du système de contrôle d'accès. Il met en évidence les acteurs impliqués et les actions



qu'ils peuvent effectuer. Le diagramme utilise un code couleur pour distinguer les différentes parties du système. Les fonctionnalités accessibles depuis l'interface graphique sont en bleu, celles liées aux QR-Codes sont en orange, et celles liées à la reconnaissance faciale sont en rouge.

À droite du diagramme, deux acteurs sont représentés : l'alimentation électrique, qui est connectée à "alimenter", et la connexion internet, qui est connectée à "Connecter au réseau". Ces acteurs sont nécessaires au bon fonctionnement du système. À gauche du diagramme, nous avons trois acteurs principaux : l'utilisateur, l'administrateur et le client. L'administrateur a plusieurs fonctionnalités à sa disposition, notamment l'accès aux systèmes embarqués des serrures, la visualisation de l'état d'occupation des ressources, la consultation de l'historique d'accès aux ressources, la gestion des droits d'accès et la gestion des pannes. La gestion des droits d'accès est étendue par la gestion des demandes des utilisateurs, la génération de QR-Codes et la gestion de la reconnaissance faciale. En cas de panne, l'administrateur peut également déverrouiller la serrure. L'acteur utilisateur est lié à plusieurs actions, dont l'authentification avec un QR-Code, l'authentification avec la reconnaissance faciale et la possibilité de réaliser une demande d'accès. Ces actions sont toutes connectées à l'action de déverrouillage de la serrure, permettant ainsi à l'utilisateur d'obtenir l'accès autorisé.

Le diagramme des cas d'utilisation permet de visualiser clairement les interactions entre les acteurs et les fonctionnalités du système. Il montre comment les utilisateurs peuvent s'authentifier, réaliser des demandes d'accès et bénéficier de la fonctionnalité de déverrouillage de la serrure.

L'administrateur dispose également d'un ensemble d'actions pour gérer les droits d'accès, surveiller l'état d'occupation des ressources et traiter les pannes éventuelles.

En résumé, ce diagramme met en évidence les principales fonctionnalités du système de contrôle d'accès, en utilisant des codes couleur pour distinguer les différentes parties du système. Il permet de comprendre de manière visuelle les interactions entre les acteurs et les actions qu'ils peuvent effectuer, facilitant ainsi la compréhension globale du fonctionnement du système.

#### h. Architecture globale du système

L'architecture globale du système repose sur un PC servant de serveur web, qui héberge l'interface de gestion, la base de données (BDD) et une imprimante. Les Raspberry Pi, connectés aux serrures QR Code et reconnaissance faciale, utilisent des caméras pour lire les QR codes et détecter les visages. Une fois l'accès authentifié, les Raspberry Pi activent les LED correspondantes et contrôlent la gâche électrique pour ouvrir la porte si l'accès est autorisé. La synchronisation des données est assurée grâce à la connexion des Raspberry Pi à la base de données (BDD) hébergée sur le PC.

Juste au-dessus de ce paragraphe, vous trouverez un schéma de montage d'une Raspberry Pi, qui représente le câblage et les connexions nécessaires pour réaliser le système. Ce schéma offre une visualisation claire de la configuration technique, facilitant ainsi la compréhension du fonctionnement global du système de contrôle d'accès.

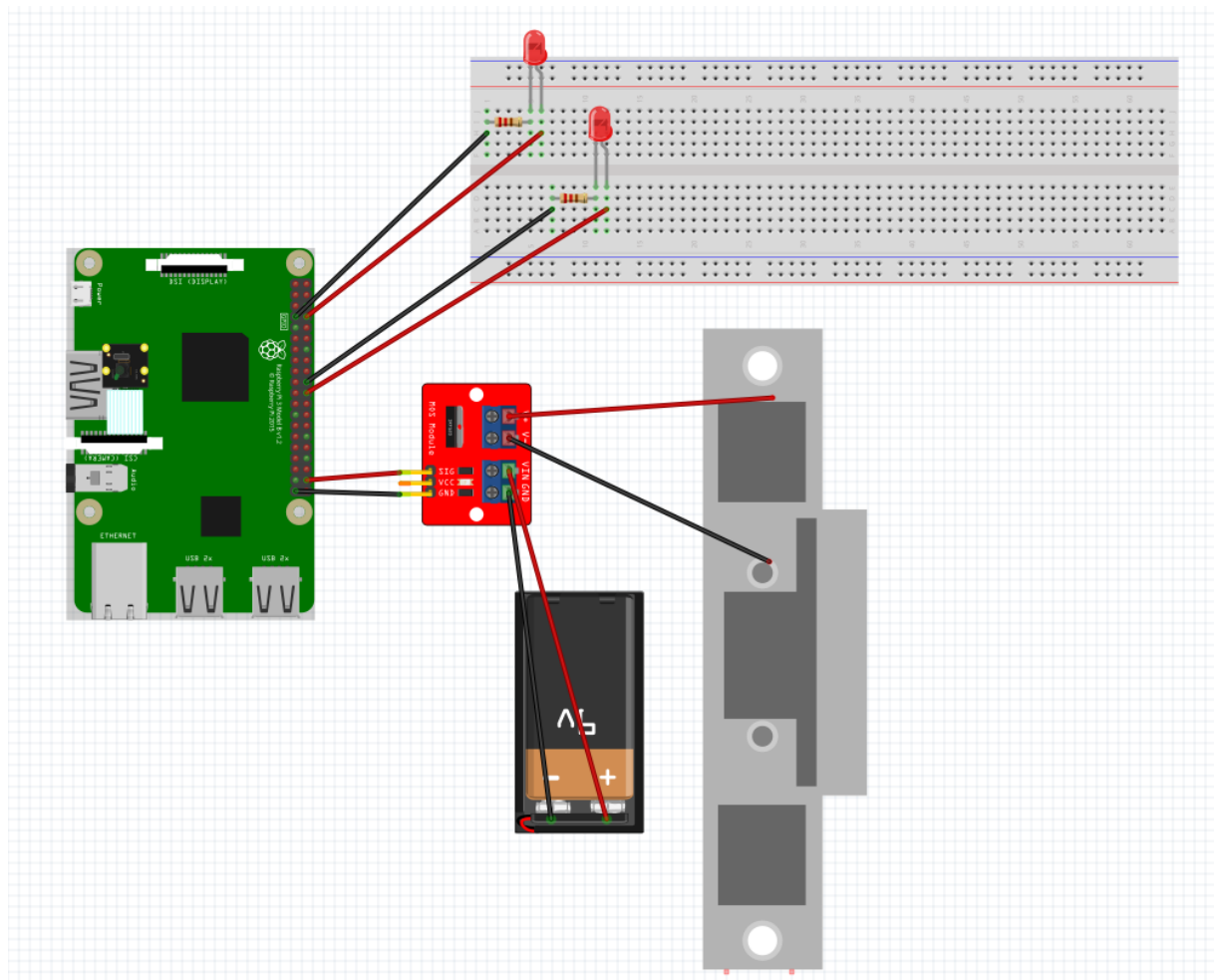


Figure 7. Schéma de montage des Raspberry Pi

Le schéma de montage du système de contrôle d'accès met en évidence les connexions et les composants utilisés pour assurer le bon fonctionnement du système. La Raspberry Pi est le cœur du système et est connectée à deux LED, une rouge et une verte. Chaque LED est alimentée par un câble d'alimentation (câble rouge) et un câble de masse (câble noir).

Ensuite, la Raspberry Pi est reliée à un transistor IRF520, dont les broches SIG et GND sont connectées à la Raspberry Pi. De l'autre côté du transistor, il est connecté à la serrure à l'aide de deux câbles. Le câble V- (câble de masse, noir) et le câble V+ (câble d'alimentation, rouge) permettent de fournir l'alimentation nécessaire à la serrure. Pour alimenter la serrure, le transistor est également connecté à VIN (câble d'alimentation, rouge) et à GND (câble de masse, noir) qui sont reliés à une pile. Cette configuration permet de contrôler l'ouverture et la fermeture de la serrure électriquement.

Grâce à cette configuration électronique, le système est en mesure de contrôler l'activation des LED (rouge et verte) pour indiquer si l'accès est autorisé ou refusé. De plus, il est capable de contrôler la gâche électrique de la serrure pour permettre l'ouverture ou la fermeture de la porte en fonction de l'autorisation d'accès.

#### i. Rôle du transistor IRF520

Le transistor IRF520 joue un rôle crucial dans le schéma de montage du système de contrôle d'accès. Sa présence est essentielle pour éviter les risques de court-circuit et protéger la Raspberry Pi.

Le transistor agit comme un interrupteur électronique qui permet de contrôler le courant entre la Raspberry Pi et la serrure électrique. Dans ce cas, le transistor est monté en tant que commutateur de type N, ce qui signifie qu'il conduit le courant lorsque le signal de commande est appliqué à sa broche SIG.

L'un des avantages clés du transistor est sa capacité à isoler électriquement les parties du système. Il divise la Raspberry Pi, qui fonctionne à une tension plus basse, de la serrure électrique, qui peut nécessiter une tension plus élevée pour son fonctionnement. En connectant la broche SIG du transistor à une broche GPIO de la Raspberry Pi, le signal de commande peut être activé ou désactivé. Lorsque le signal de commande est appliqué, le transistor conduit le courant et permet à la Raspberry Pi de contrôler l'alimentation de la serrure électrique. Cela permet à la Raspberry Pi de décider quand activer ou désactiver la serrure.

En évitant un contact direct entre la Raspberry Pi et la serrure, le transistor offre une protection supplémentaire à la Raspberry Pi contre les variations de tension potentielles qui pourraient endommager le circuit ou compromettre son fonctionnement. Il agit comme une barrière électrique en fournissant un chemin de courant contrôlé entre la Raspberry Pi et la serrure. Grâce à la présence du transistor, la Raspberry Pi peut communiquer de manière sûre et fiable avec la serrure électrique, en fournissant les signaux de commande nécessaires pour l'ouverture et la fermeture de la porte. Cette fonctionnalité garantit un fonctionnement stable du système de contrôle d'accès et prolonge la durée de vie de la Raspberry Pi en la protégeant contre d'éventuels dommages.

En résumé, le transistor IRF520 joue un rôle essentiel dans le schéma de montage en fournissant une isolation électrique entre la Raspberry Pi et la serrure électrique. Il agit comme un interrupteur électronique contrôlé par la Raspberry Pi, permettant de gérer l'alimentation de la serrure de manière sécurisée et fiable. Sa présence protège la Raspberry Pi contre les risques de court-circuit et garantit un fonctionnement optimal du système de contrôle d'accès.

### **III. Conception de la base de données**

#### **j. Le rôle de la base de données synchronisée**

La base de données synchronisée joue un rôle essentiel dans l'authentification fiable des utilisateurs dans notre système de contrôle d'accès pour le tribunal de grande instance d'Evry.

L'authentification des utilisateurs est une étape critique pour garantir la sécurité du système et des ressources sensibles du tribunal. En centralisant les données d'authentification dans une base de données synchronisée, nous pouvons mettre en place des mécanismes solides pour vérifier l'identité des utilisateurs de manière fiable. La base de données stocke les informations d'identification des utilisateurs autorisés, telles que leurs données personnelles, leurs codes d'accès, et les informations biométriques dans le cas de la reconnaissance faciale. Lorsqu'un utilisateur tente d'accéder aux ressources, l'interface de gestion peut interroger la base de données pour authentifier l'utilisateur en comparant les données fournies avec celles enregistrées.

Grâce à cette authentification fiable basée sur la base de données synchronisée, nous pouvons éviter les accès non autorisés et prévenir les tentatives de contournement du système. Les informations d'authentification sont mises à jour en temps réel dans la base de données, ce qui garantit que seules les personnes autorisées peuvent accéder aux ressources et aux locaux sensibles du tribunal. De plus, la synchronisation de la base de données permet également de maintenir l'historique des accès, ce qui facilite la traçabilité et la vérification ultérieure des activités des utilisateurs. En enregistrant les données d'accès, telles que la date, l'heure et les ressources consultées, dans la base de données synchronisée, nous pouvons créer un journal d'audit complet et fiable pour des raisons de sécurité et de responsabilité.

En résumé, la base de données synchronisée joue un rôle crucial dans l'authentification fiable des utilisateurs dans notre système de contrôle d'accès. Elle stocke les informations d'identification, permettant une vérification précise de l'identité des utilisateurs autorisés, et facilite la traçabilité des activités à des fins d'audit et de sécurité. Grâce à cette synchronisation, nous garantissons un niveau élevé de sécurité et de confiance dans l'accès aux ressources du tribunal de grande instance d'Evry.

#### **k. Hébergement de la base de données**

Lorsque j'ai conçu la base de données pour gérer les demandes d'accès et les autorisations au tribunal de grande instance d'Evry, il était essentiel de trouver un moyen fiable et accessible d'héberger la base de données. Après mûre réflexion, j'ai opté pour l'utilisation de XAMPP, un environnement de développement web polyvalent, qui m'a permis de créer un serveur local pour héberger la base de données.

XAMPP est une suite logicielle qui combine Apache, MySQL, PHP et phpMyAdmin. J'ai choisi d'installer XAMPP sur mon ordinateur principal, qui agit comme un serveur pour le projet. Cela m'a offert plusieurs avantages. Tout d'abord, en utilisant Apache comme serveur web, j'ai pu accéder à ma base de données et à d'autres fichiers liés au projet, tels que l'interface de gestion, à partir de n'importe quel navigateur web sur mon réseau local. Cela signifie que mes camarades de projet et moi-même pouvions accéder à la base de données via nos Raspberry Pi en utilisant l'adresse IP de mon ordinateur principal.

En ce qui concerne la gestion de la base de données, j'ai utilisé phpMyAdmin, une interface conviviale qui permet d'administrer facilement la base de données MySQL. Avec phpMyAdmin, j'ai pu créer et gérer les tables, exécuter des requêtes SQL et effectuer d'autres tâches d'administration essentielles. J'ai également créé des identifiants d'accès spécifiques pour mes camarades de projet, leur permettant ainsi d'accéder à la base de données avec leurs propres Raspberry Pi.



Figure 8. Logo de XAMPP, phpMyAdmin et Apache

## I. Processus de conception

Lorsque j'ai entrepris de concevoir une base de données pour gérer les demandes d'accès et les autorisations au tribunal de grande instance d'Evry, j'ai fait preuve de créativité et de réflexion approfondie. À l'époque, je cherchais une approche simple et intuitive pour répondre aux besoins de base de ce système.

Pour concevoir la structure de la base de données, j'ai commencé par analyser les différentes entités impliquées. J'ai identifié les ressources, les demandes d'accès et les relations entre elles comme les composants clés du système. Ensuite, j'ai créé la table "Ressources" pour stocker les informations relatives aux ressources, telles que le nom, le type de ressource, la sensibilité, le statut et la capacité. Cependant, avec le recul, j'ai réalisé que l'ancienne version de la base de données présentait quelques lacunes. L'une des erreurs que j'ai commises était de ne pas avoir pris en compte la possibilité d'une capacité variable pour chaque ressource. J'avais utilisé une seule colonne "Capacité" pour stocker la capacité de toutes les ressources, ce qui limitait la flexibilité du système.

De plus, j'ai constaté que la gestion des demandes d'accès était insuffisante dans l'ancienne version. La table "Demandes\_accès" stockait les informations basiques des demandes, comme le nom, le prénom, l'e-mail, les dates et le statut. Cependant, il manquait des fonctionnalités importantes, telles que la génération de QR codes uniques pour chaque demande, ce qui aurait renforcé la sécurité du système.

Pour corriger ces erreurs et améliorer la conception de la base de données, j'ai effectué des recherches approfondies et trouvé de l'inspiration dans d'autres systèmes de contrôle d'accès complexes. J'ai réalisé que j'avais besoin d'une approche plus robuste pour gérer les relations entre les demandes d'accès, les ressources et les visages des utilisateurs.

Dans la nouvelle version de la base de données, j'ai consciemment amélioré la structure en introduisant de nouvelles tables et relations pour permettre une gestion plus précise et flexible des informations. Une des améliorations majeures a été l'ajout de la table "Visages" qui permet de stocker les informations relatives aux visages des utilisateurs, ainsi que leur statut de

validation. De plus, j'ai créé les tables intermédiaires "Demandes\_acces\_ressources" et "Visages\_acces\_ressources" pour gérer les relations many-to-many entre les demandes, les visages et les ressources. Ces tables de liaison permettent de représenter de manière efficace les associations complexes entre ces entités. Elles jouent un rôle essentiel dans la gestion des accès en permettant de déterminer rapidement quelles demandes ont accès à quelles ressources et quels visages sont associés à quelles ressources. La table "Demandes\_acces\_ressources" permet de relier les demandes d'accès aux ressources correspondantes, tandis que la table "Visages\_acces\_ressources" établit les liens entre les visages et les ressources. Grâce à ces relations many-to-many, il devient possible de gérer facilement les scénarios où plusieurs demandes peuvent avoir accès à plusieurs ressources et où plusieurs visages peuvent être associés à différentes ressources.

Ces améliorations ont permis de prendre en compte la capacité variable des ressources en ajoutant les colonnes "Capacite\_totale" et "Capacite\_actuelle" dans la table "Ressources". De plus, j'ai introduit la génération de QR codes uniques pour chaque demande en ajoutant la colonne "Valeur\_QR\_Code" dans la table "Demandes\_acces".

En résumé, l'ancienne version de la base de données, bien que réalisée avec créativité et réflexion, présentait des lacunes en termes de gestion de la capacité des ressources et de fonctionnalités de demande d'accès. Cependant, grâce à des recherches approfondies et à l'inspiration tirée d'autres systèmes de contrôle d'accès, j'ai pu concevoir une nouvelle version de la base de données qui corrigeait ces erreurs et offrait une structure plus solide et flexible pour gérer les demandes d'accès et les autorisations au tribunal de grande instance d'Evry.

Pour mieux illustrer ces changements, voici deux schémas : un représentant l'ancienne version et un représentant la version finale.

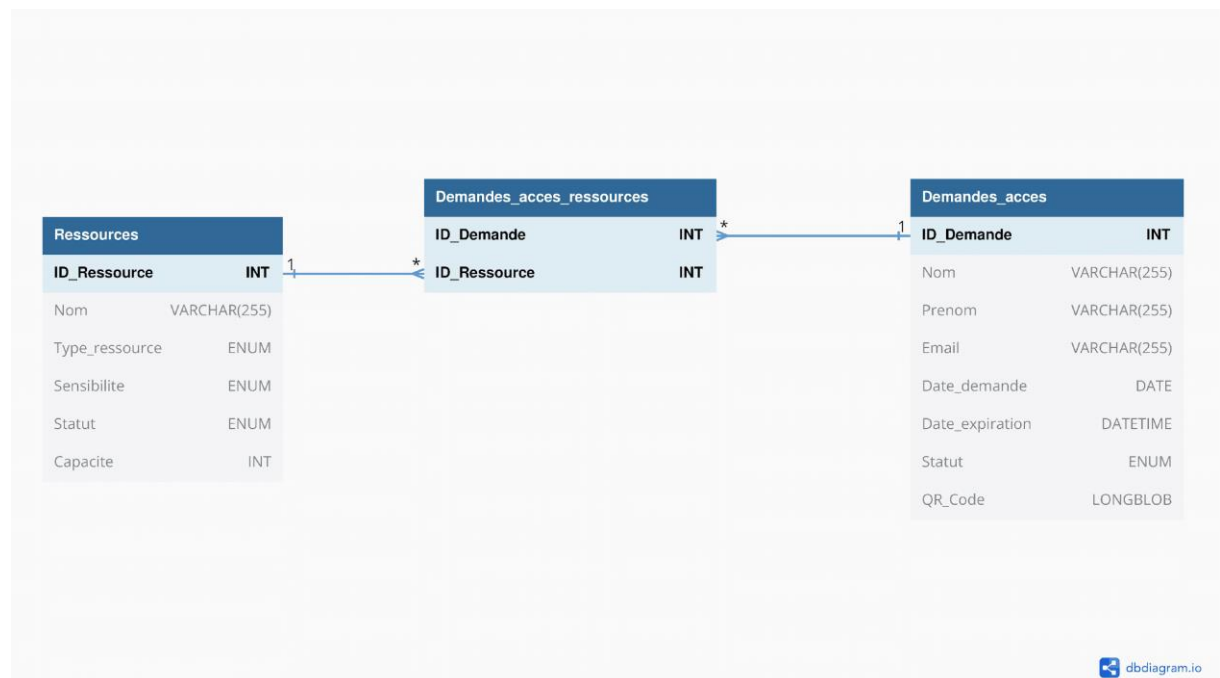
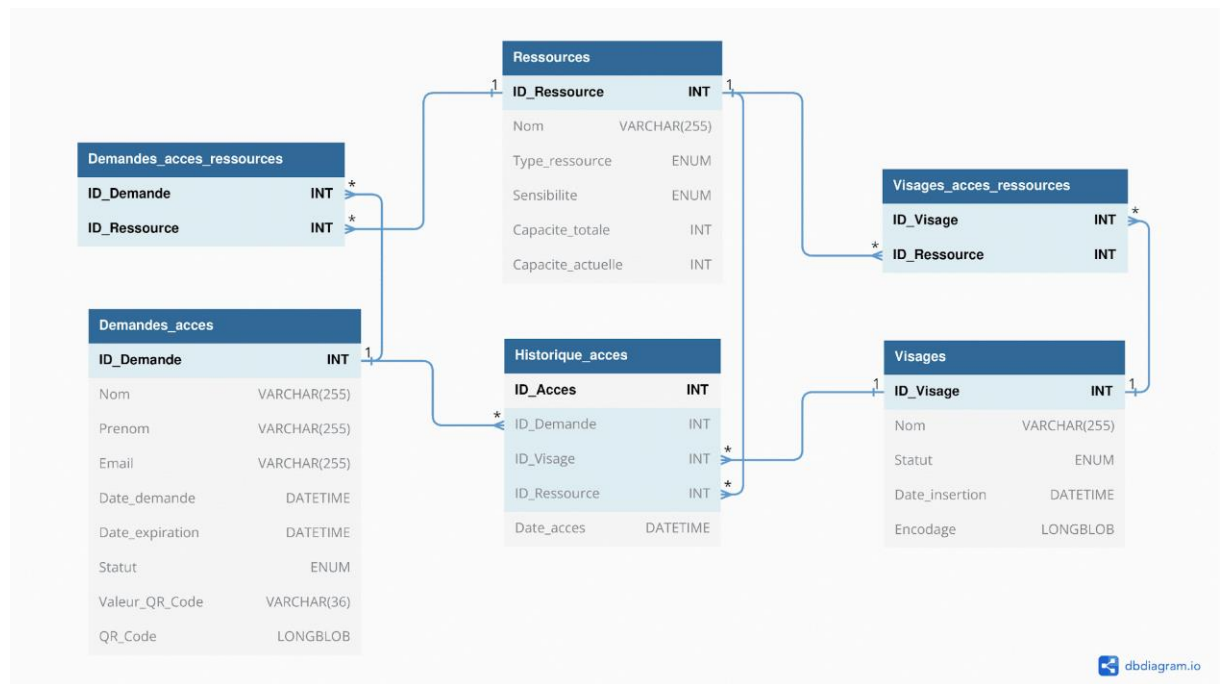


Figure 9. (a) première version de la base de données



(b) version finale de la base de données

#### m. Evènements phpmyadmin

Pour assurer une mise à jour constante des données dans la base de données, j'ai intégré des événements dans ma base de données gérée par phpMyAdmin. Les événements sont des fonctionnalités qui permettent d'automatiser des tâches récurrentes et périodiques. Ils jouent un rôle essentiel dans la gestion et la mise à jour automatique des données. Dans le contexte de mon projet, j'ai utilisé les événements pour effectuer des mises à jour régulières et cohérentes, assurant ainsi la fiabilité et l'intégrité de ma base de données.

- **Évènement 1 : Mise à jour du statut des demandes expirées toutes les 20 secondes :**  
L'évènement "Mise à jour du statut des demandes expirées" a été créé pour garantir que les demandes qui ont atteint leur date d'expiration soient correctement traitées. Toutes les 20 secondes, l'évènement se déclenche et exécute une requête de mise à jour. Cette requête vérifie si la date d'expiration d'une demande est antérieure ou égale à l'instant présent (NOW()). Si c'est le cas, le statut de la demande est automatiquement modifié pour passer à "Expiré". Cela permet de maintenir la base de données à jour et de refléter avec précision l'état actuel des demandes.

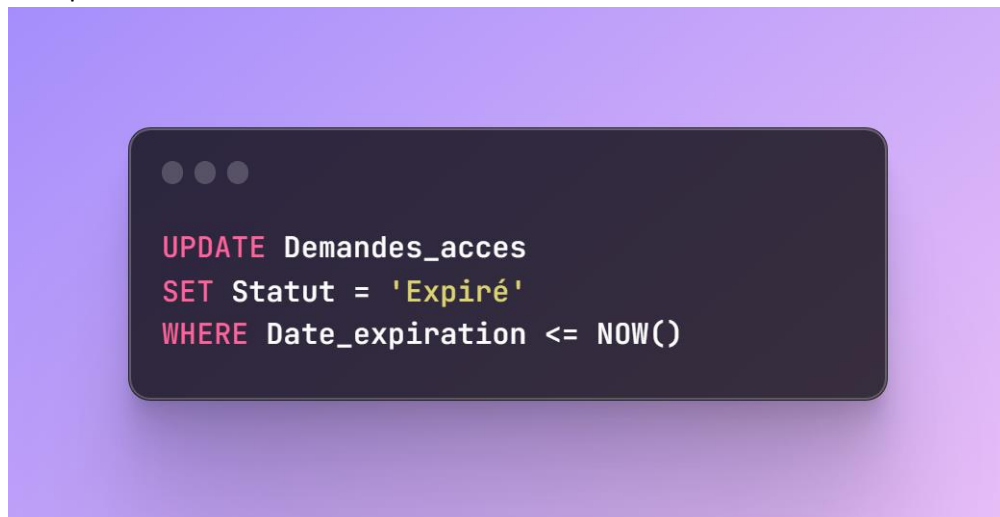



Figure 10. (a) Evènement 1 de la base de données



- **Événement 2 : Mise à jour des capacités des ressources toutes les 30 secondes** : Pour assurer une gestion efficace des capacités des ressources, j'ai mis en place l'événement "Mise à jour des capacités des ressources". Cet événement se déclenche toutes les 30 secondes et exécute une requête de mise à jour. La requête calcule la capacité actuelle des ressources en soustrayant le nombre total de demandes validées associées à chaque ressource de la capacité totale de cette dernière. Cette mise à jour régulière des capacités garantit que les informations affichées dans la base de données sont constamment à jour, offrant ainsi une vision précise des ressources disponibles.



```
UPDATE Ressources r
SET Capacite_actuelle = r.Capacite_totale - (
SELECT COUNT(*) FROM Demandes_acces d
INNER JOIN Demandes_acces_ressources dr ON d.Id_demande = dr.Id_demande
WHERE dr.Id_ressource = r.Id_ressource
AND d.Statut = 'Validé')
```

(b) Événement 2 de la base de données

Ces événements automatisés dans ma base de données contribuent grandement à l'efficacité et à la précision des opérations de gestion de mon projet. Ils permettent de maintenir les données à jour, d'optimiser les processus et de garantir un fonctionnement fluide de ma base de données.

## IV. Conception de l'interface de gestion (IG)

### n. Fonctionnalités de l'interface de gestion

L'interface de gestion que j'ai développée pour le système Contrôle-Accès du tribunal de grande instance d'Evry est un élément clé de ce projet ambitieux. Permettez-moi de vous expliquer le processus de conception que j'ai suivi pour créer cette interface intuitive et fonctionnelle.

L'interface de gestion, comme la base de données, est hébergée sur XAMPP, une plateforme de développement web, dans le répertoire "htdocs". Cela permet d'accéder à l'interface à partir d'un navigateur web sur le même ordinateur où XAMPP est installé. Cette configuration offre une grande flexibilité et facilite la mise en place du système de contrôle d'accès.

Pour commencer, j'ai réalisé un croquis détaillé de l'interface que je souhaitais mettre en place. J'ai envisagé une barre de navigation fixe en haut de l'écran, offrant un accès facile à toutes les pages du système. Au centre de la page, j'ai prévu un encadré où serait affiché le contenu correspondant à la page sélectionnée. Ce choix de conception visait à assurer une navigation fluide et une expérience utilisateur agréable.

En ce qui concerne la réalisation de l'interface, j'ai adopté une approche progressive. J'ai commencé par développer toutes les fonctionnalités principales en utilisant HTML, PHP et SQL, sans me soucier de l'aspect visuel final. Cette approche m'a permis de me concentrer sur la logique et le bon fonctionnement du système. Une fois que toutes les fonctionnalités ont été implémentées et que le système était pleinement opérationnel, j'ai consacré du temps à améliorer l'aspect esthétique de l'interface. J'ai ajouté du CSS (Cascading Style Sheets) pour embellir les éléments visuels, harmoniser les couleurs, les polices et les espaces, et rendre l'interface plus agréable à utiliser.

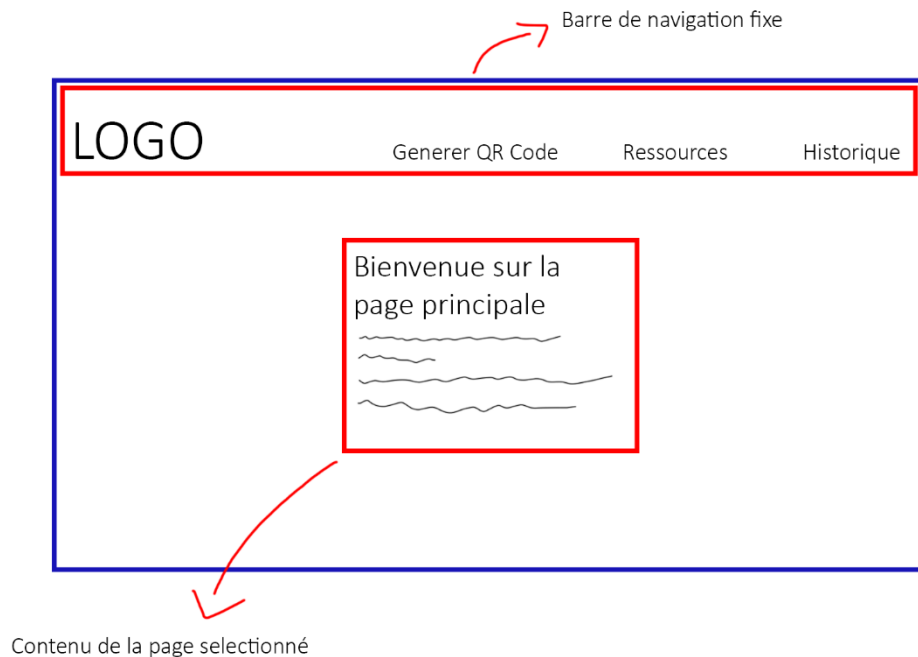


Figure 11. Croquis de l'interface graphique

En conclusion, l'interface de gestion que j'ai développée pour le système Contrôle-Accès du tribunal de grande instance d'Evry offre un large éventail de fonctionnalités visant à simplifier et à optimiser la gestion des accès aux ressources. Voici un récapitulatif des principales fonctionnalités de cette interface graphique :

- **Validation des demandes d'accès des utilisateurs** : L'interface permet de consulter et de valider les demandes d'accès soumises par les utilisateurs, offrant un contrôle précis sur les autorisations accordées.
- **Génération de codes d'accès sous forme de QR codes** : Grâce à cette fonctionnalité, l'interface permet de générer des codes d'accès temporaires ou permanents sous forme de QR codes, facilitant ainsi l'accès aux ressources autorisées.
- **Gestion de l'accès par reconnaissance faciale** : L'interface prend en charge la gestion de l'accès aux locaux sensibles grâce à la reconnaissance faciale, garantissant une sécurité renforcée.
- **Messagerie électronique** : Elle intègre également un système messagerie électronique qui facilite la communication lors de la génération des codes d'accès sous forme de QR codes. Grâce à cette fonctionnalité, les utilisateurs peuvent recevoir des notifications par e-mail contenant les informations relatives à leur demande d'accès.
- **Visualisation de l'état d'occupation des ressources** : L'interface fournit une vue globale et synthétique sur le niveau d'occupation des ressources, offrant ainsi une meilleure gestion et planification des accès.
- **Consultation de l'historique d'accès aux ressources** : Grâce à cette fonctionnalité, les gestionnaires peuvent accéder à l'historique des accès aux ressources en temps réel, permettant une traçabilité complète des activités.
- **Accès aux systèmes embarqués des serrures** : L'interface offre la possibilité d'accéder aux systèmes embarqués des serrures, permettant ainsi la gestion et le contrôle à distance de ces dispositifs.

En combinant ces différentes fonctionnalités, l'interface de gestion offre un environnement convivial et intuitif pour administrer et contrôler efficacement l'accès aux ressources du tribunal de grande instance d'Evry, contribuant ainsi à une gestion simplifiée, une sécurité renforcée et une meilleure utilisation des ressources disponibles.

## o. Demandes d'accès

La fonctionnalité de gestion des demandes d'accès permet de gérer efficacement les demandes de codes QR. J'ai identifié deux possibilités pour générer ces codes QR.

- **Génération des codes d'accès par le personnel d'accueil** : Le personnel d'accueil, depuis la page "Génération des codes d'accès", peut générer les codes QR en utilisant le formulaire présent sur la page. Le formulaire, développé en HTML et PHP, permet de collecter les informations nécessaires telles que le nom, le prénom, l'e-mail et la date d'expiration (si applicable) pour le code QR. Une fois que le personnel d'accueil remplit les champs requis et coche les ressources auxquelles donner accès, il clique sur le bouton "Générer QR Code".

Lorsque le bouton est cliqué, les données saisies dans le formulaire sont envoyées au serveur web, où le code PHP correspondant traite les informations. Le code PHP se connecte à la base de données hébergée sur XAMPP, dans le répertoire htdocs, en utilisant les informations d'identification appropriées. Il effectue les opérations nécessaires pour générer le code QR correspondant à la demande d'accès. Les informations de la demande, y compris les données personnelles de l'utilisateur, la date d'expiration et les ressources sélectionnées, sont enregistrées dans la base de données.

Une fois le code QR généré, il est affiché à l'écran dans l'interface de gestion pour que le personnel d'accueil puisse le visualiser. Le code QR est ensuite envoyé à l'utilisateur par e-mail, en utilisant une fonctionnalité que nous détaillerons dans une section ultérieure.

The screenshot shows a web interface for 'Tribunal d'Evry'. At the top, there is a navigation bar with links: 'Generer un QR Code', 'Gestion des demandes', 'Gestion Visages', 'Historique accès', 'Ressources', and 'Accès aux serrures'. The main content area features a form titled 'Génération des codes d'accès'. The form includes input fields for 'Nom', 'Prénom', and 'Email', each with a placeholder text 'Entrez votre [nom/prénom/email]'. There is also a 'Date d'expiration' field with a date picker showing 'jj/mm/aaaa --:--'. Below these fields is a section for 'Ressources' with several checkboxes: 'Salle audience 1', 'Salle audience 2', 'Salle de délibération 1', 'Salle des pas perdus', 'Salle de conférence', and 'Salle de réunion 1'. At the bottom of the form is a large blue button labeled 'GÉNÉRER QR CODE'.

Figure 12. (a) Page acces\_admin.php

- **Envoi de la demande par les utilisateurs** : Les utilisateurs ont également la possibilité de soumettre leur demande en utilisant une page dédiée. Sur cette page, ils saisissent leur nom, prénom, e-mail et éventuellement une date d'expiration pour le code QR. Après avoir rempli le formulaire, les utilisateurs cliquent sur le bouton "Envoyer la demande".

Lorsque le bouton est cliqué, les données du formulaire sont envoyées au serveur web, où le code PHP correspondant traite la demande. Tout comme dans le cas précédent, le code PHP se connecte à la base de données hébergée sur XAMPP et enregistre les informations de la demande. La demande est ensuite visible dans la page "Gestion des demandes", à laquelle le personnel d'accueil a accès. Ils peuvent consulter toutes les informations de la demande.

**Tribunal d'Evry**

### Demande d'accès aux ressources

Nom :

Prénom :

Email :

Date d'expiration :

Ressources :

- ☐ Salle audience 1
- ☐ Salle audience 2
- ☐ Salle de délibération 1
- ☐ Salle des pas perdus
- ☐ Salle de conférence
- ☐ Salle de réunion 1

**ENVOYER LA DEMANDE**

(b) Page acces\_utilisateur.php

Le personnel d'accueil dispose de fonctionnalités pour gérer ces demandes dans l'interface de gestion :

1. **Accepter une demande** : Si le personnel d'accueil décide d'accepter une demande, ils peuvent marquer la demande comme "Acceptée" dans l'interface de gestion. Lorsqu'ils cliquent sur le bouton d'acceptation, un e-mail contenant le code QR correspondant est automatiquement envoyé à l'utilisateur à l'adresse e-mail fournie dans la demande. Le statut de la demande est alors mis à jour pour refléter son approbation.
2. **Refuser une demande** : Si une demande ne peut pas être acceptée, le personnel d'accueil a la possibilité de la refuser. En cliquant sur le bouton de refus, le statut de la demande est mis à jour pour indiquer qu'elle a été "Refusée". Dans ce cas, aucun e-mail contenant un code QR n'est envoyé à l'utilisateur.

ID	Nom	Prénom	Email	Date de la demande	Date d'expiration	Statut	Ressources	QR Code	Action
4	Galara	Jaques	venerezbusiness@gmail.com	2023-05-29 17:41:04	2023-05-31 22:41:00	En attente	Salle audience 2 Salle de délibération 1 Salle de réunion 1	-	<div>Accepter</div> <div>Refuser</div>
3	Galara	Jaques	venerezbusiness@gmail.com	2023-05-29 17:40:42	2023-05-30 17:46:00	Validé	Salle audience 1 Salle de délibération 1 Salle des pas perdus Salle de réunion 1	Voir le QR Code	-
2	Omar	Hussein	omarhussein3366@gmail.com	2023-04-08 13:21:15	2023-04-08 15:19:00	Validé	Salle audience 1 Salle audience 2 Salle de délibération 1 Salle des pas perdus Salle de conférence Salle de réunion 1	Voir le QR Code	-
1	Omar	Hussein	omarhussein3366@gmail.com	2023-04-08 13:19:55	2023-04-08 15:19:00	Validé	Salle audience 1 Salle audience 2 Salle de délibération 1	Voir le QR Code	-

(c) Page gestion\_demandes\_utilisateur.php

Cette approche offre une solution complète pour la gestion des demandes d'accès, en permettant au personnel d'accueil de générer des codes QR pour les utilisateurs en remplissant un formulaire intuitif et en offrant aux utilisateurs la possibilité de soumettre leurs demandes directement. Elle garantit également la sécurité et la confidentialité des informations personnelles tout au long du processus.

p. Envoi des courriels électroniques

L'interface de gestion de l'application utilise le service de messagerie Sendinblue et la bibliothèque PHPMailer pour l'envoi des courriels électroniques.

- Sendinblue est un service de messagerie en ligne qui offre des fonctionnalités d'envoi d'e-mails avancées. Il agit en tant que serveur SMTP (Simple Mail Transfer Protocol) pour notre application, permettant ainsi l'envoi sécurisé et fiable des courriels. Sendinblue offre une infrastructure robuste et une interface conviviale pour la configuration des paramètres d'envoi d'e-mail, tels que les informations d'authentification SMTP, le type de sécurité et le port.
- PHPMailer, quant à lui, est une bibliothèque PHP populaire et puissante qui facilite l'envoi d'e-mails à partir d'applications PHP. Il fournit une interface simple et conviviale pour la création, la personnalisation et l'envoi d'e-mails. PHPMailer offre des fonctionnalités avancées telles que l'envoi d'e-mails en format HTML, la gestion des pièces jointes, la gestion des destinataires multiples et la configuration des paramètres SMTP. Il est largement utilisé dans le développement Web pour faciliter l'envoi d'e-mails personnalisés et automatisés.



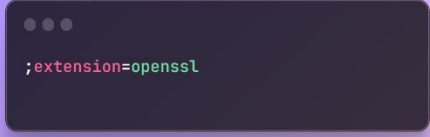
Figure 13. (a) Logo de Sendinblue

(b) Logo de PHPMailer

Pour utiliser PHPMailer avec XAMPP, il est nécessaire d'activer l'extension OpenSSL dans la configuration de PHP. OpenSSL est une bibliothèque open-source utilisée pour les communications sécurisées via des protocoles tels que HTTPS, SMTPS, etc. L'activation de cette extension est essentielle pour permettre à PHPMailer d'établir une connexion sécurisée avec le serveur SMTP

Voici les étapes pour activer l'extension OpenSSL sur XAMPP :

1. Localisez le dossier d'installation de XAMPP sur votre système. Par défaut, il est généralement installé dans le répertoire C:\xampp (sur Windows) ou /Applications/XAMPP (sur macOS)
2. Ouvrez le fichier php.ini avec un éditeur de texte
3. Recherchez la ligne suivante dans le fichier 'php.ini' :

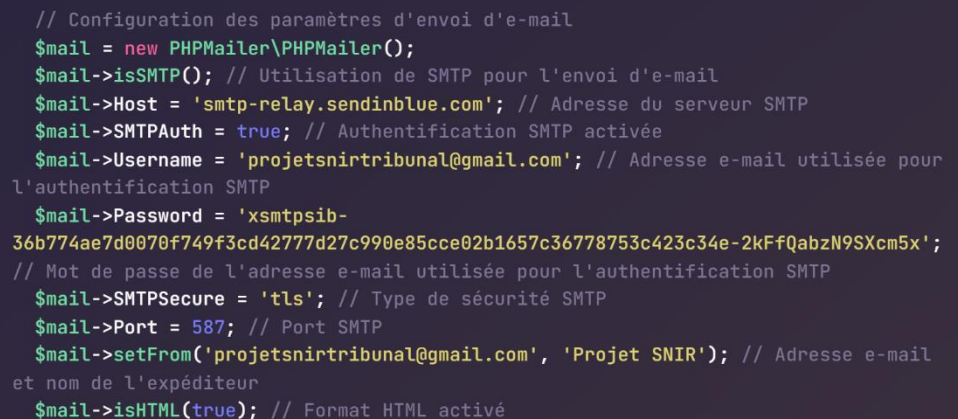


```
;extension=openssl
```

4. Retirez le point-virgule (;) en début de ligne pour décommenter l'extension OpenSSL et enregistrez.

Voici les détails techniques de l'implémentation pour l'envoi des courriels électroniques avec SendinBlue et PHPMailer :

1. Configuration des paramètres d'envoi d'e-mail : La configuration des paramètres d'envoi d'e-mail se fait dans le code PHP. La bibliothèque PHPMailer est utilisée pour se connecter au serveur SMTP de Sendinblue. Les paramètres d'authentification tels que l'adresse e-mail de l'expéditeur, le nom d'utilisateur, le mot de passe, le type de sécurité et le port SMTP sont spécifiés dans le code.



```
// Configuration des paramètres d'envoi d'e-mail
$mail = new PHPMailer\PHPMailer();
$mail->isSMTP(); // Utilisation de SMTP pour l'envoi d'e-mail
$mail->Host = 'smtp-relay.sendinblue.com'; // Adresse du serveur SMTP
$mail->SMTPAuth = true; // Authentification SMTP activée
$mail->Username = 'projetsnirtribunal@gmail.com'; // Adresse e-mail utilisée pour
l'authentification SMTP
$mail->Password = 'xsmtpsib-
36b774ae7d0070f749f3cd42777d27c990e85cce02b1657c36778753c423c34e-2kFfQabzN9SXcm5x';
// Mot de passe de l'adresse e-mail utilisée pour l'authentification SMTP
$mail->SMTPSecure = 'tls'; // Type de sécurité SMTP
$mail->Port = 587; // Port SMTP
$mail->setFrom('projetsnirtribunal@gmail.com', 'Projet SNIR'); // Adresse e-mail
et nom de l'expéditeur
$mail->isHTML(true); // Format HTML activé
```



2. Envoi de l'e-mail de confirmation : Lorsque la demande d'accès est validée, un e-mail de confirmation est envoyé à l'adresse e-mail fournie. Le contenu de l'e-mail est personnalisé et contient un message de confirmation ainsi que le code QR en tant que pièce jointe (Le fonctionnement détaillé de la generation du QR Code sera exploré en détail dans une section ultérieure du rapport).

```
$mail->addAddress($email);  
$mail->CharSet = 'UTF-8';  
$mail->Subject = 'Votre demande d\'accès a été validée';  
$mail->Body = 'Bonjour '.$prenom.',<br><br>Votre demande d\'accès a été validée.<br><br>Vous trouverez ci-joint votre QR Code qui vous permettra d\'accéder aux<br><br>ressources demandés.<br><br>Cordialement<br><br>Le Tribunal d\'Evry';  
$mail->addAttachment("qr_codes/$nom-$prenom.png");  
$mail->send();
```

#### q. Génération des QR-Codes

Dans notre projet, j'ai utilisé la bibliothèque PHPQRCode pour générer les QR codes. Cette bibliothèque m'a permis de simplifier la génération des QR codes uniques et personnalisés associés à chaque demande d'accès.

- **PHP QR Code (phpqrcode)** est une bibliothèque open source largement utilisée pour la génération de codes QR en PHP. Cette bibliothèque offre des fonctionnalités puissantes et faciles à utiliser pour créer des codes QR personnalisés. Elle permet de générer des QR codes à partir de divers types de données, tels que du texte, des URL, des informations de contact, des numéros de téléphone, des adresses email, etc. Grâce à phpqrcode, il est possible de personnaliser l'apparence des QR codes en ajustant leur taille, leur niveau de correction d'erreurs, leurs couleurs et même en ajoutant un logo. Cette bibliothèque est bien documentée, dispose d'une communauté active et offre une solution fiable pour la génération de QR codes dans nos projets PHP

Pour pouvoir utiliser cette bibliothèque, il est nécessaire d'activer l'extension GD (Graphics Draw) dans le serveur web local tel que XAMPP. L'extension GD permet la manipulation d'images, y compris la création de codes QR à l'aide de PHPQRCode :

1. Ouvrez le fichier "php.ini" de votre installation XAMPP
2. Recherchez la ligne qui commence par ";extension=gd" et supprimez le point-virgule (;) au début de la ligne pour activer l'extension GD
3. Enregistrez le fichier "php.ini" et Redémarrez le serveur Apache de XAMPP

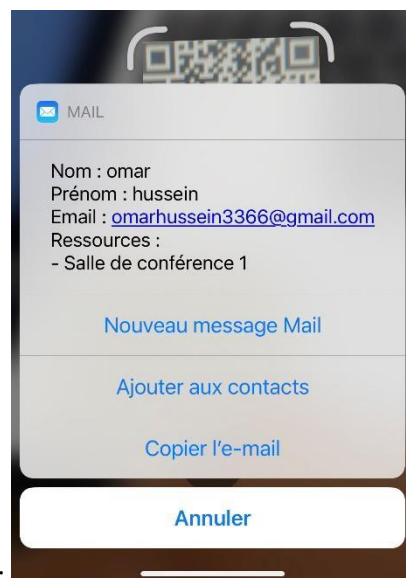


Figure 14. Logo de PHP Qr Code

Dans la version précédente de mon code, je construisais manuellement les données à encoder dans le QR code en utilisant une approche basée sur des chaînes de caractères. Je rassemblais les informations telles que le nom, le prénom, l'email, la date d'expiration et les ressources sélectionnées pour créer une chaîne de caractères représentant le contenu du QR code. Ensuite, j'utilisais la fonction `QRcode::png()` de la bibliothèque pour générer le QR code à partir de ces données et je le stockais localement dans le répertoire "qr\_codes" de mon serveur XAMPP

```
// Construction de l'URL qui sera encodée dans le QR code
$data = "Nom : $nom\nPrénom : $prenom\nEmail : $email\nExpiration:
$expiration\nRessources :\n";
foreach ($ressources as $ressource) {
    $query = "SELECT Nom FROM ressources WHERE ID_Ressource = ?";
    $stmt = mysqli_prepare($link, $query);
    mysqli_stmt_bind_param($stmt, "i", $ressource);
    mysqli_stmt_execute($stmt);
    $result = mysqli_stmt_get_result($stmt);
    $row = mysqli_fetch_array($result);
    $data .= "- " . $row['Nom'] . "\n";
}

// Génération du QR code
QRcode::png($data, "qr_codes/$nom-$prenom.png");
```



Voici le rendu une fois le QR-Code généré et scanné :

Cependant, j'ai réalisé que cette approche comportait des risques de sécurité et de maintenabilité. Les informations personnelles, comme l'email, étaient directement incluses dans le contenu du QR code, ce qui pouvait compromettre la confidentialité des données sensibles. De plus, la construction manuelle des données à encoder était sujette à des erreurs de format et rendait la gestion des caractères spéciaux plus complexe.

Dans la version améliorée de mon code, j'ai donc décidé d'utiliser la bibliothèque PHP QR Code pour simplifier la génération des QR codes et renforcer la sécurité des données. Maintenant, je génère le contenu du QR code en utilisant une valeur aléatoire unique, générée par la fonction `bin2hex(random_bytes(18))`. Cette valeur est stockée dans la variable `$valeur_qr_code`.

Ensuite, j'utilise la fonction `QRcode::png()` pour générer le QR code à partir de la valeur `$valeur_qr_code`. Le QR code ainsi créé est ensuite enregistré localement dans le répertoire "qr\_codes" de mon serveur XAMPP.

Il est également important de noter que j'enregistre le QR code généré dans la base de données. J'utilise la fonction `file_get_contents()` pour récupérer le contenu du fichier QR code généré, puis j'utilise la fonction `mysqli_real_escape_string()` pour échapper les caractères spéciaux dans le contenu du QR code. Enfin, j'exécute une requête SQL pour mettre à jour la table `Demandes_acces` avec le QR code correspondant à l'ID de la demande d'accès.

```
// Construction de l'URL qui sera encodée dans le QR code
$data = "$valeur_qr_code";

// Génération du QR code
QRcode::png($data, "qr_codes/$nom-$prenom.png");

// Enregistrement du QR code dans la base de données
$qr_code = file_get_contents("qr_codes/$nom-$prenom.png");
$qr_code = mysqli_real_escape_string($link, $qr_code);
$query = "UPDATE Demandes_acces SET QR_Code = '$qr_code' WHERE ID_Demande = $id_demande";
mysqli_query($link, $query);
```



Voici le rendu une fois le QR-Code généré et scanné :

Cette approche améliorée de génération et de gestion des QR codes offre une meilleure sécurité des données et simplifie la manipulation des informations personnelles. Les QR codes sont stockés à la fois localement sur notre serveur XAMPP et dans la base de données, ce qui permet un accès sécurisé et une traçabilité des demandes d'accès.

r. Visualisation de l'état d'occupation des ressources

Je souhaite maintenant vous présenter la fonctionnalité de visualisation de l'état d'occupation des ressources que j'ai développée dans le cadre de mon projet. J'ai créé une page simple qui affiche les données stockées dans la base de données. Grâce à un événement que j'ai précédemment expliqué dans le rapport, l'attribut "Capacité\_actuelle" des ressources se met automatiquement à jour, ce qui permet d'avoir une vue en temps réel de leur occupation.



Tribunal d'Evry					
Generer un QR Code   Gestion des demandes   Gestion Visages   Historique accès   Ressources   Accès aux serrures					
ID Ressource	Nom	Type	Sensibilité	Capacité totale	Capacité actuelle
1	Salle audience 1	Salle audience	Sensible	50	50
2	Salle audience 2	Salle audience	Sensible	75	75
3	Salle de délibération 1	Salle de délibération	Sensible	20	20
4	Salle des pas perdus	Salle des pas perdus	Non Sensible	200	200
5	Salle de conférence	Salle de conférence	Sensible	100	100
6	Salle de réunion 1	Salle de réunion	Non Sensible	10	10

Figure 15. Page etat\_occupation\_ressources.php

Lorsque l'utilisateur accède à cette page, une connexion à la base de données est établie. Ensuite, une requête SQL est exécutée pour récupérer les données sur les ressources à partir de la table "Ressources". Une fois les données obtenues, elles sont affichées dans un tableau à l'aide de balises HTML. Chaque ligne du tableau représente une ressource et les colonnes affichent différentes informations telles que l'identifiant de la ressource, son nom, son type, sa sensibilité, sa capacité totale et sa capacité actuelle.

```
<table class="tableau2">
  <thead>
    <tr>
      <th>ID Ressource</th>
      <th>Nom</th>
      <th>Type</th>
      <th>Sensibilité</th>
      <th>Capacité totale</th>
      <th>Capacité actuelle</th>
    </tr>
  </thead>
  <tbody>
    <?php while ($row = mysqli_fetch_array($result)): ?>
      <tr>
        <td><?php echo $row['ID_Ressource']; ?></td>
        <td><?php echo $row['Nom']; ?></td>
        <td><?php echo $row['Type_ressource']; ?></td>
        <td><?php echo $row['Sensibilite']; ?></td>
        <td><?php echo $row['Capacite_totale']; ?></td>
        <td><?php echo $row['Capacite_actuelle']; ?></td>
      </tr>
    <?php endwhile; ?>
  </tbody>
</table>
```

#### s. Historique d'accès aux ressources

La page "Historique d'accès" fonctionne de manière similaire à celle de l'état des ressources, mais avec une différence clé. Cette page récupère les données de la table "Historique\_acces" de la base de données. L'objectif est de fournir un historique détaillé des accès effectués dans le système.



ID_Acces	Nom Demande	Nom Visage	Ressource	Date accès
1		Omar Hussein	Salle audience 2	2023-04-08 13:14:39
2		Omar Hussein	Salle audience 2	2023-04-08 13:14:40
11	Omar Hussein		Salle des pas perdus	2023-04-08 13:33:47
12	Omar Hussein		Salle des pas perdus	2023-04-08 13:33:47

Figure 16. Page historique.php

Pour mettre à jour cette page, les Raspberry Pi relié aux serrures sont utilisés pour enregistrer les connexions et les accès. Le fonctionnement de l'enregistrement des accès, sera détaillé ultérieurement dans les parties "Serrure QR-Code" et "Serrure reconnaissance faciale" du rapport.

- Pour la serrure QR-Code, lorsqu'un utilisateur présente son QR-Code généré à partir de la page "Générer un QR-Code", le Raspberry Pi connecté à la serrure scanne le code et envoie une requête à la base de données pour vérifier les informations associées au QR-Code. Si les données correspondent et que l'accès est autorisé, un enregistrement d'accès est créé dans la table "Historique\_acces".
- Quant à la serrure reconnaissance faciale, lorsque l'utilisateur se place devant le système de reconnaissance faciale connecté au Raspberry Pi, celui-ci capture l'image du visage et l'envoie pour un processus de reconnaissance. Si le visage est reconnu avec succès et que l'accès est autorisé, un enregistrement d'accès est également créé dans la table "Historique\_acces".

Dans les parties "Serrure QR-Code" et "Serrure reconnaissance faciale" du rapport, les détails techniques de chaque méthode seront abordés en détail. Cela inclura les algorithmes utilisés, les librairies ou technologies spécifiques, ainsi que les mesures de sécurité mises en place pour garantir l'intégrité du processus d'enregistrement des accès

#### t. Gestion de l'accès par reconnaissance faciale

Sur la page de gestion de l'accès par reconnaissance faciale, j'ai développé une interface conviviale pour gérer les visages enregistrés dans la table "Visages" de la base de données. Cette page permet aux administrateurs d'afficher les informations associées à chaque visage, y compris l'ID du visage, le nom de la personne, le statut actuel, la date d'insertion et la possibilité de visualiser l'image du visage et d'effectuer une action.

ID Visage	Nom	Statut	Date insertion	Image	Gestion de l'accès	Action
1	Elon Musk	Révoqué	2023-04-08 13:14:20	Afficher l'image	Gérer l'accès	Redonner l'accès
2	Ryan Reynolds	Validé	2023-04-08 13:14:20	Afficher l'image	Gérer l'accès	Révoquer l'accès
4	Sfera Ebbasta	Validé	2023-04-08 13:14:20	Afficher l'image	Gérer l'accès	Révoquer l'accès
5	Omar Hussein	Validé	2023-04-08 13:14:20	Afficher l'image	Gérer l'accès	Révoquer l'accès
6	Stephen Hawking	Validé	2023-05-25 01:02:32	Afficher l'image	Gérer l'accès	Révoquer l'accès

Figure 17. Page liste\_visages.php

En parcourant les enregistrements de visages, j'ai inclus un bouton permettant de visualiser l'image du visage, ce qui facilite l'identification visuelle de la personne concernée. Cela offre une visualisation effective du visage et aide les administrateurs à prendre des décisions éclairées.

```
// Récupération de l'image à partir de la base de données
$sql = "SELECT * FROM Visages WHERE ID_Visage = $id";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    $image = $row["Encodage"];
    header("Content-type: image");
    echo $image;
} else {
    echo "Image non trouvée";
}
```

Ce code récupère une image à partir de la base de données en fonction d'un ID de visage spécifique. Si l'image est trouvée, elle est affichée dans le navigateur ; sinon, un message d'erreur est affiché.

Une autre colonne importante dans cette page est "Gestion de l'accès". Lorsque l'administrateur clique sur ce bouton, il est redirigé vers une autre page qui affiche toutes les ressources disponibles dans le système. Cette page offre la possibilité de sélectionner les ressources auxquelles la personne associée au visage enregistré doit avoir accès. L'administrateur peut simplement cocher les ressources souhaitées pour donner l'autorisation d'accès à ces ressources spécifiques.



Figure 18. Page `gerer_acces_visages.php`

Les ressources auxquelles le visage a déjà accès sont préalablement cochées. Lorsque l'utilisateur soumet le formulaire en cliquant sur le bouton "Valider", les ressources sélectionnées sont enregistrées dans la base de données.

Le code de la page commence par établir une connexion à la base de données. Ensuite, il récupère l'ID du visage sélectionné à partir de la page précédente et le nom correspondant à cet ID. Ensuite, il récupère toutes les ressources disponibles et les ressources auxquelles le visage a déjà accès. Le code génère ensuite le formulaire avec des cases à cocher pour chaque ressource, en cochant les cases appropriées en fonction de l'accès actuel du visage. Lorsque le formulaire est soumis, les ressources sélectionnées sont envoyées à la page "enregistrer\_acces\_visages.php" pour être enregistrées dans la base de données.

Enfin, la dernière colonne est intitulée "Action". Cette colonne affiche des boutons d'action qui dépendent du statut actuel du visage enregistré. Si le statut est "Validé", un bouton est affiché pour révoquer l'accès, permettant ainsi à l'administrateur de retirer l'autorisation d'accès pour ce visage spécifique. D'un autre côté, si le statut est "Révoqué", un bouton est affiché pour redonner l'accès, donnant ainsi la possibilité à l'administrateur de rétablir l'autorisation d'accès pour ce visage.



## u. Accès aux serrures

Sur la page "Accès aux serrures", vous trouverez toutes les informations nécessaires pour vous connecter aux systèmes embarqués des serrures du tribunal. Cette page a été conçue pour faciliter l'accès et la gestion des serrures en fournissant des détails importants et les identifiants de connexion requis.

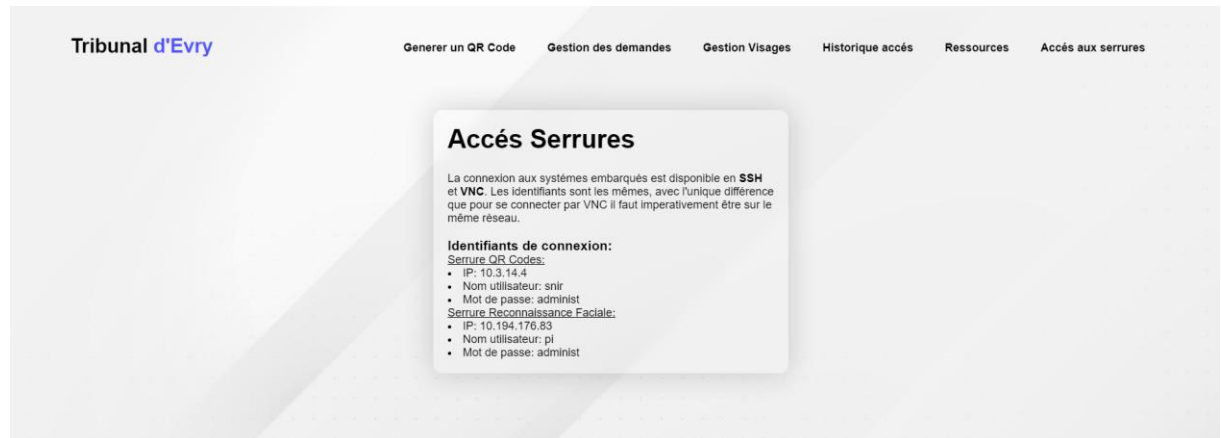


Figure 19. Page acces\_serrures.html

- La première solution est l'utilisation de VNC (Virtual Network Computing), qui permet d'accéder aux interfaces graphiques des systèmes embarqués des serrures. L'avantage de VNC est qu'il offre une expérience utilisateur conviviale et intuitive. En utilisant VNC, vous pourrez interagir visuellement avec les serrures, ce qui facilite la configuration et la gestion des paramètres.
- La deuxième solution est l'utilisation de SSH (Secure Shell), un protocole sécurisé pour établir une connexion à distance avec les serrures. SSH offre des mesures de sécurité avancées, notamment le cryptage des données et l'authentification à clé publique/privée, ce qui garantit la confidentialité et l'intégrité des communications. L'utilisation de SSH vous permettra de gérer les serrures en ligne de commande, en exécutant des commandes et des scripts pour effectuer diverses opérations.

En choisissant ces deux méthodes de connexion, j'ai cherché à offrir une flexibilité maximale aux utilisateurs. La connexion par VNC permet une interaction graphique directe, tandis que la connexion par SSH offre un accès sécurisé en ligne de commande. Ainsi, l'utilisateur peut choisir la méthode qui convient le mieux à ses besoins et préférences.

## v. Conception visuelle : L'optimisation de l'interface grâce au CSS

Le CSS joue un rôle crucial dans la conception de l'interface en définissant l'apparence visuelle des éléments. Il permet de contrôler la taille, la couleur, la typographie, les marges, les espacements, les bordures, les arrière-plans, les effets de transition et bien plus encore. En combinant différentes propriétés CSS, il est possible de créer des designs uniques et attrayants, qui reflètent l'identité visuelle d'une entreprise ou d'une marque. Grâce au CSS, il est également possible de structurer les pages web de manière cohérente et organisée.

Ci-dessus, je vous présente l'avant et l'après de l'implémentation du CSS dans la page de génération des codes d'accès, mettant en évidence la différence frappante dans l'apparence et le style des pages web avant et après l'utilisation du CSS.

### Tribunal d'Evry

- [Générer un QR Code](#)
- [Gestion des demandes](#)
- [Gestion Visages](#)
- [Historique accès](#)
- [Ressources](#)
- [Accès aux serrures](#)

#### Génération des codes d'accès

Nom :

Prénom :

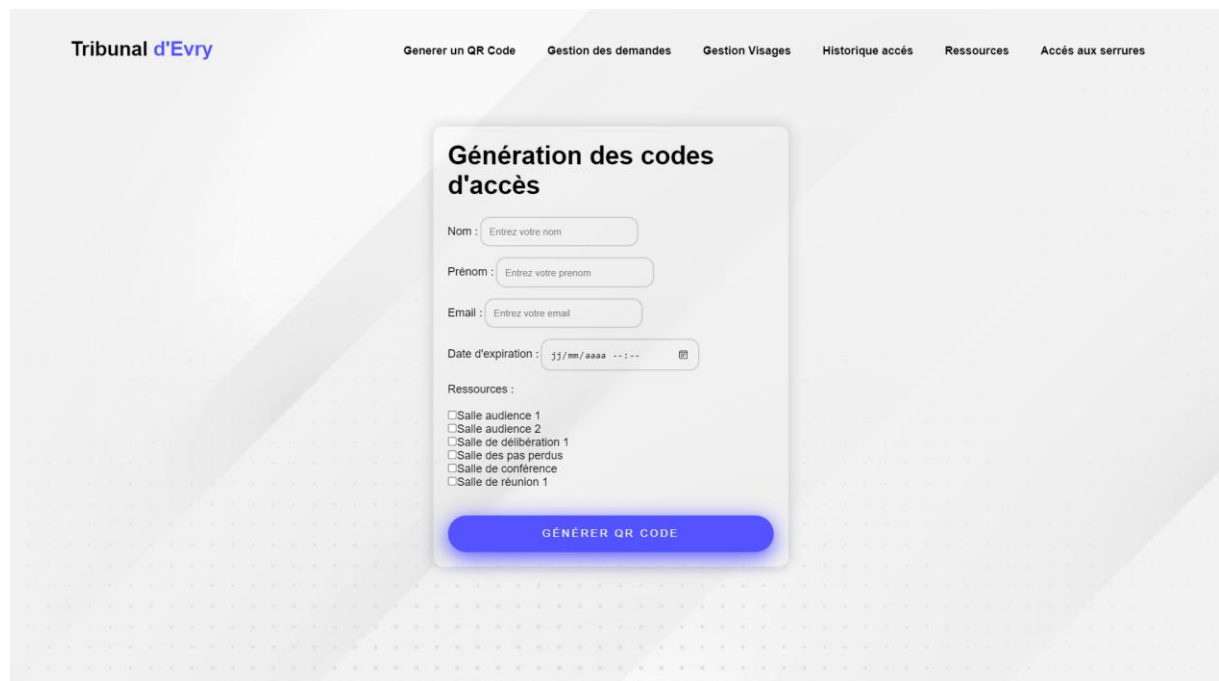
Email :

Date d'expiration :  

Ressources :

- ☐ Salle audience 1
- ☐ Salle audience 2
- ☐ Salle de délibération 1
- ☐ Salle des pas perdus
- ☐ Salle de conférence
- ☐ Salle de réunion 1

Figure 20. (a) Page de génération des codes d'accès avant CSS



Tribunal d'Evry


Générer un QR Code Gestion des demandes Gestion Visages Historique accès Ressources Accès aux serrures

### Génération des codes d'accès

Nom :

Prénom :

Email :

Date d'expiration :  

Ressources :

- ☐ Salle audience 1
- ☐ Salle audience 2
- ☐ Salle de délibération 1
- ☐ Salle des pas perdus
- ☐ Salle de conférence
- ☐ Salle de réunion 1

**GÉNÉRER QR CODE**

(b) Page de génération des codes d'accès après CSS

Avant l'implémentation du CSS, les pages web étaient dépourvues de style et d'esthétique, mais grâce à son utilisation, l'interface graphique a été transformée, offrant une expérience visuelle attrayante et professionnelle.

Dans du développement CSS, j'ai opté pour l'utilisation de l'outil uiverse.io pour la partie CSS. Je souhaite mettre en valeur ce choix, car il s'est avéré être une ressource extrêmement précieuse dans le processus de conception et de stylisation de l'interface graphique.

Uiverse.io m'a permis d'accéder à une vaste collection d'éléments d'interface utilisateur préconstruits, tels que des boutons, des formulaires, des cartes et bien d'autres encore. Ces éléments étaient non seulement visuellement attrayants, mais ils étaient également faciles à intégrer dans mon projet. Le fait qu'ils soient open source et gratuits à utiliser a été un avantage majeur, me permettant d'économiser du temps et des ressources.

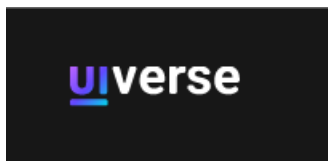


Figure 21. Logo uiverse.io

Ce choix m'a permis d'accélérer considérablement le processus de développement, en me permettant de me concentrer sur d'autres aspects cruciaux de mon projet. L'intégration fluide des éléments de l'interface utilisateur fournis par uiverse.io a contribué à créer une expérience utilisateur attrayante et très professionnelle.

Voici l'explication des parties les plus importantes du code CSS :

- **Sélection globale (\*) :**

```
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
}
```

Cette partie du code applique des styles à tous les éléments de la page. Les propriétés `margin: 0;` et `padding: 0;` suppriment les marges et les espaces de remplissage par défaut des éléments, assurant une mise en page cohérente. La propriété `box-sizing: border-box;` garantit que la largeur et la hauteur des éléments incluent également leur bordure et leur rembourrage.

- **Styling des formulaires :**

```
.formulaire {
  width: 500px;
  padding: 20px;
  border-radius: 12px;
  backdrop-filter: blur(8px);
  box-shadow: 0 0px 40px #23232333;
  transition: 0.2s ease-out;
  margin: 4% 0% 0% 35%;
}

.formulaire:hover {
  box-shadow: 0 0 20px #23232333;
}

.formulaire h1 {
  font-size: 36px;
  margin-bottom: 25px;
}
```

Cette partie du code style un formulaire spécifique avec la classe `.formulaire`. Il définit la largeur, le rembourrage et le rayon de bordure, créant une boîte bien définie pour le formulaire. La propriété `backdrop-filter: blur(8px);` ajoute un flou d'arrière-plan au formulaire, lui donnant une apparence plus esthétique. Le `box-shadow` crée une ombre légère autour du formulaire. Lorsque vous survolez le formulaire, l'ombre est réduite grâce à la pseudo-classe `:hover`. De plus, la taille de la police et les marges pour le titre `<h1>` à l'intérieur du formulaire sont spécifiées.

- **Bouton de validation :**

```
.generer{
  width: 100%;
  padding: 17px 40px;
  border-radius: 50px;
  border: 0;
  background-color: rgb(255, 255, 255);
  box-shadow: rgb(0 0 0 / 5%) 0 0 8px;
  letter-spacing: 1.5px;
  text-transform: uppercase;
  font-size: 15px;
  transition: all .5s ease;
  cursor: pointer;
}

.generer:hover{
  letter-spacing: 3px;
  background-color: rgb(83, 83, 255);
  color: hsl(0, 0%, 100%);
  box-shadow: rgb(83, 83, 255) 0px 7px 29px 0px;
}

.generer:active{
  letter-spacing: 3px;
  background-color: rgb(83, 83, 255);
  color: hsl(0, 0%, 100%);
  box-shadow: rgb(83, 83, 255) 0px 0px 0px 0px;
  transform: translateY(10px);
  transition: 100ms;
}
```

Ce code spécifie que le bouton doit avoir une largeur de 100% de son conteneur parent, un padding de 17 pixels en haut et en bas, ainsi que de 40 pixels à gauche et à droite. Les bordures sont supprimées et les coins du bouton sont arrondis avec un rayon de 50 pixels, lui donnant une forme circulaire. La couleur de fond est définie comme blanc, avec une légère ombre pour créer un effet en relief. L'espacement entre les lettres est augmenté de 1,5 pixels pour une meilleure lisibilité. Lorsque le curseur survole le bouton, la couleur de fond change en bleu avec une transition en douceur, la lettre-spacing est augmentée à 3 pixels et une nouvelle ombre est ajoutée pour renforcer l'effet de surbrillance. Lorsque le bouton est activé, il est enfoncé de 10 pixels vers le bas avec une transition plus rapide.