

Institut National de Radioélectricité et de Cinématographie

Enseignement technique secondaire de qualification

Accès aux études supérieures



Avenue Jupiter, 188

1190 Forest

Coffret de sûreté

Projet personnel de Michalowski Alan

Pour l'obtention du certificat de qualification

Technicien(ne) en informatique

Année scolaire : 2025 – 2026

1. Remerciement

Mes remerciements s'adressent à mon père, pour son savoir en menuiserie qui a permis la création des maquettes de mon projet. Son soutien technique m'a permis de régler les problèmes matériels liées au travail du bois.

Contents

1.	Remerciement.....	1
2.	Introduction	4
2.1	Contexte du projet	4
2.2	Problématique.....	4
2.3	Objectifs du TFE	4
2.4	Présentation rapide de la solution.....	4
3.	Analyse des besoins	5
3.1	Description du problème	5
3.2	Public cible et utilisateurs	5
3.3	Cahier des charges.....	5
3.4	Fonctionnalités attendues	6
3.5	Contraintes du projet	6
4.	Étude préalable (recherche).....	7
4.1	Solutions existantes	7
4.2	Comparaison.....	7
4.3	Choix technologiques.....	7
4.4	Justification des choix	8
5.	Conception du projet	11
5.1	Architecture globale	11
5.2	Modélisation de la base de données	12
5.3	Maquettes et Interfaces Utilisateurs	12
5.4	Organisation des fichiers et structure du code	17
6.	Réalisation (développement).....	18
6.1	Explication du fonctionnement	18
6.2	Présentation des fonctionnalités implémentées.....	18
6.3	Extraits de code commentés	20
6.4	Difficultés rencontrées	31
6.5	Solutions apportées	32
7.	Tests, Débogage et Validations	32
7.1	Protocoles de tests réalisés.....	32
7.2	Résultats obtenus.....	33
7.3	Bugs rencontrés	33

7.4	Corrections	33
8	Conclusion.....	34
8.1	Résumé du travail réalisé	34
8.2	Objectifs atteints ou non.....	34
8.3	Compétences acquises.....	34
8.4	Améliorations possibles	34
9.	Bibliographie / Webographie	35
10.	Annexes.....	37

2. Introduction

2.1 Contexte du projet

Ce projet s'inscrit dans le cadre de ma dernière année d'étude et a pour ambition de réaliser un projet concret des différentes matières d'options abordées durant tout mon parcours scolaire à l'Inraci en option informatique. La conception d'un système de coffrets de sûreté me permet d'assembler des compétences accomplies durant les cours d'informatique embarquée, de structure matérielle et logistique, ainsi que réseau et des systèmes d'exploitation. Le choix de la sécurité s'est imposé naturellement car c'est un domaine où l'innovation technologique n'a pas de limite et est primordiale. Le projet contient des éléments du projet de ma 5^{ème} années.

2.2 Problématique

Comment garantir une sécurité totale et sans erreur pour le stockage d'objets de valeur, tout en offrant une gestion fluide et moderne pour l'utilisateur ? La problématique se base sur l'équilibre entre une protection physique (contre les intrusions et les catastrophes naturelles), une accessibilité numérique via des applications de gestion à distance, sans jamais mettre en danger les biens de clients et le confort du client.

2.3 Objectifs du TFE

L'objectif principal de ce travail est de concevoir un écosystème complet de sécurisation, allant de la gestion administrative à la protection physique. Les objectifs spécifiques incluent la conception technique d'une salle sécurisée intelligente capable de réagir aux différents types de risques (gaz, inondations, vibrations). Le développement d'un site web permettant la location et la gestion des abonnements selon la taille des coffrets et la durée de location. L'implémentation d'un protocole de double authentification pour sécuriser l'accès physique aux coffrets. La réalisation d'une maquette fonctionnelle représentant le bâtiment pour démontrer la faisabilité du concept. La création d'un environnement de sécurité et d'aide pour les employeurs.

2.4 Présentation rapide de la solution

La solution proposée repose sur une liaison entre la surveillance humaine et l'automatisation technologique. Elle comprend une salle protégée par un réseau de capteurs environnementaux et de caméras, dont l'accès est régulé par un garde équipé d'une interface de gestion centralisée. Côté client, l'expérience est digitalisée car le client loue son coffret via un site internet dédié ou directement à la réception. Pour garantir un niveau de sécurité de haut qualité, l'accès physique

au coffret impose une triple identification : une validation par le garde sur la tablette, l'usage d'un badge personnel et la saisie d'un code PIN de sécurité. Enfin, une zone logistique est prévue pour les livreurs, garantissant la protection des colis, même en l'absence du propriétaire.

3. Analyse des besoins

3.1 Description du problème

À une époque où la sécurité des biens matériels est une préoccupation croissante, de nombreuses personnes cherchent des solutions fiables pour protéger leurs objets de valeur, objets sentimentaux ou leurs documents importants. De l'autre côté, le problème du commerce en ligne a fait apparaître une nouvelle problématique comme le vol, la dégradation ou l'abandon de colis lors de l'absence des destinataires. L'enjeu de ce projet est de concevoir un système capable de répondre à ces deux besoins. Il s'agit de créer une infrastructure de haute sécurité, qui est facile en utilisation pour les clients alliant une surveillance physique, une liaison de donnée automatique et une gestion numérique moderne.

3.2 Public cible et utilisateurs

Le projet s'adresse à un public large et varié comme les particuliers souhaitant mettre à l'abri des objets précieux dans un environnement plus sûr qu'un domicile privé. Les utilisateurs qui utilisent des services de livraison souvent absentes à leurs domiciles et craignant que leurs colis ne soient laissés sans surveillance par les livreurs. Tous les âges confondus grâce à son utilité simple et varié.

3.3 Cahier des charges

Afin de mieux réaliser le projet j'ai décidé de faire un cahier des charges, qui permet de voir tous les besoins. La pièce doit être protégée contre les intrusions, mais aussi contre les risques naturels comme inondation, gaz, vibrations cela permet d'avoir un accès environnementale sécurisé. La double authentification est l'accès à un coffret qui ne peut se faire que par la validation d'un garde de sécurité via la tablette de sécurité, de l'utilisateur via un badge RFID et un code pin de 4 chiffres pour que le voleur du badge RFID n'ouvre toujours pas le coffret. Chaque événement (ouverture, alerte, connexion) doit être enregistré dans un système de logs pour garantir un historique complet en cas de vol.

3.4 Fonctionnalités attendues

Afin que mon projet possède un ordre de réalisation, je l'ai séparé en plusieurs parties afin de bien comprendre les différentes fonctionnalités, de bien les distinguer et localiser des éventuelles erreurs. D'abord je commence avec le site web où les clients peuvent créer un compte, se connecter, modifier leurs données en cas d'erreurs, consulter des offres, réaliser des paiements sécurisés des abonnements, voir des informations sur notre entreprise. Une base de données qui contient toutes les informations sur les employés, les clients, les coffrets, les logs et l'admin. L'application d'employé est une interface permettant au personnel de se connecter, gérer les comptes clients, d'attribuer des coffrets, de vérifier les paiements, d'avoir accès aux logs et faire une création d'un compte employé par l'admin qui est le patron de l'entreprise. Système de sécurité contrôlé par une application qui est sur un Raspberry Pi 4 qui pilote les capteurs, gère les caméras en direct, contrôle les servomoteurs pour l'ouverture des portes et coffrets, l'activation ou désactivation d'alarme de sécurité. Pour la fin une création d'une maquette du bâtiment et du coffret pour bien représenter le fonctionnement du projet. En théorie, lorsqu'un client s'inscrit, paie son abonnement via le site web et met un code PIN, tout s'enregistre dans la base de données. Pour cette étape il y a une deuxième possibilité car si un client ne sait pas utiliser un site web ou faire des paiements en lignes, par exemple une personne âgée, elle peut aller se rendre à la réception où un employeur créera le compte à sa place avec un mot de passe temporaire grâce à l'application d'employé. Quand le client a créé son compte et a payé un abonnement en ligne, il doit se rendre à la réception pour une vérification de données, une attribution d'un badge RFID. Quand cela est fait il a le droit de rentrer dans la pièce des coffrets de sûreté, si un garde de sécurité est présent, il peut lui ouvrir la porte car il a vu sur les caméras et dans les logs qu'il a été se présenter à la réception. Si le garde de sécurité n'est pas là, le client peut mettre son PIN sur la key pad devant la porte. Quand il rentre dans la pièce la personne doit mettre le code PIN, doit mettre son badge et avoir l'autorisation de la personne de sécurité pour ouvrir son coffret. Quand un livreur arrive avec des colis, il doit donner les noms et prénoms des clients concernés, il reçoit une liste de numéro de coffret avec le nom pour qu'il place les colis dans les bons coffrets. Quand il mettra le colis dans le coffret, le client recevra une notification que son colis est prêt à récupérer.

3.5 Contraintes du projet

La principale contrainte est la nécessité de faire communiquer de manière fluide une base de données avec le site web, l'application des employés, l'application de sécurité. Cela fait un réel défi car tout doit être lié ensemble et fonctionner ensemble. En tant qu'entreprise de sûreté,

le système ne peut tolérer aucune erreur logicielle ou matérielle qui pourrait créer le danger de la sécurité des biens de clients. Les scénarios d'essais de vols et de hack sont énormément nombreuses et je dois assurer que chaque tentative sera bloquée.

4. Étude préalable (recherche)

4.1 Solutions existantes

Avant de débiter, j'ai analysé les solutions de stockage et de sécurisation actuellement disponibles sur le marché. J'ai trouvé Les coffres-forts bancaires qui offrent une sécurité maximale. Mais sont coûteux, faut avoir un compte bancaire chez la banque et n'offrent pas de gestion numérique moderne pour le client car tout se fait par rendez-vous. Comme une deuxième option, il y a des casiers à colis qui sont très pratiques pour la logistique et envoi une notification quand le colis arrive. Mais leur niveau de sécurité est insuffisant pour des objets de très haute valeur et elles ne possèdent pas de capteurs environnementaux.

4.2 Comparaison

Quand on regarde la situation actuelle sur le marché, il n'y a pas de solutions qui propose ces deux options en même temps comme le stockage d'objets de valeurs et la récupération de colis. En reliant ces deux types de stockage on a des coffrets de stockage d'objets de valeurs, des casiers de récupération de colis avec l'envoi de message et une sécurité maximale. En plus de ça, je rajoute un prix agréable, une disponibilité pour chaque client, une gestion numérique pour les clients.

4.3 Choix technologiques

En utilisant mes compétences acquises en cours d'option et en effectuant des recherches sur internet, je devais prendre des choix sur le fonctionnement technologique de mon projet. :

- Hébergement & Web, Paiement : D'abord je devais garantir que le client puisse créer son compte sur mon site web, pour faire ceci j'ai choisi l'hébergeur LWS. Pour la création du site web j'ai opté pour les langages HTML, CSS, Javascript et PHP. Afin de garantir un moyen de paiement fiable et sécuriser, j'ai utilisé Stripe.
- Environnements de développement : Pour un codage sans aucune perturbation j'ai choisi Visual Studio Code comme mon IDE principale et j'ai choisi Filezilla pour la connexion FTP à mon hébergeur LWS et Always j'ai choisi Filezilla.

- Base de données : Pour la sauvegarde de tous mes clients, employés, coffrets, logs, j'ai décidé d'utiliser Always Data qui utilise le langage MySQL.
- Développement Application & Matériel : Pour la création de l'application j'ai choisi Python avec ses bibliothèques Tkinter, mysql-connector-python, bcrypt, pyinstaller. Un Raspberry pi 4.

4.4 Justification des choix

LWS : Une des premières consignes était d'héberger un nom de domaine pour avoir une affiche de son projet TFE, j'ai profité pour en plus de l'affiche faire tout un système de création de comptes clients et d'acheter un abonnement pour un coffret, pour faire ceci j'ai choisi l'hébergeur LWS car il propose des abonnements avec des prix différents avec des fonctions en moins, et non un abonnement avec toutes les fonctionnalités qui auraient monté le prix.

HTML : Le langage HTML est un langage de programmation qui permet de créer la base du site web et structurer des éléments.

CSS : Le langage CSS permet de transformer les pages HTML visuellement pour obtenir une apparence plus belle pour l'œil du client.

Javascript : Le langage Javascript permet de vérifier si les demandes nécessaires sont appliquées. J'ai décidé de l'utiliser pour que le client remplit tous les champs demander et pour vérifier s'il respecte les demandes nécessaires comme @ dans le mail ou 8 caractères 1 majuscule, 1 chiffre dans le mot de passe.

PHP : Le langage PHP permet de faire la liaison entre le site web et la base de données. Pour n'importe quelle création, connexion, modification, paiement, ... Il faut faire des imports et des exports vers la base de données afin de tout sauvegarder et d'utiliser ces sauvegardes.

Stripe : Stripe est un moyen de paiements en légal en Belgique, d'abord je voulais crée mon propre moyen de paiement mais cela est interdit, Stripe permet d'enregistrer tous les paiements réaliser sur mon site web, en offrant une sécurité génial parfait pour mon projet. Il est très simple en utilisation comme la création de produit, voir les paiements des clients en détails, possède un moyen de remboursement, offre des options de paiement unique et mensuel parfait pour les abonnements. Le système de Webhooks permet de communiquer directement avec mon script PHP grâce aux clés id uniques pour confirmer qu'un paiement a été réussi, déclenchant ainsi automatiquement l'attribution du coffret au client dans le système.

Visual Studio Code : VS Code est un IDE qui offre plusieurs langages de programmation parfait pour moi car j'en utilise plusieurs HTML, CSS, ... Je voulais utiliser Pycharm car c'est mon IDE de confiance et il ne fait pas d'erreurs de reconnaissance de langages automatique comme parfois VS Code mais il est utilisable que pour le langage Python.

Always Data : J'ai choisi Always Data pour sa fiabilité dans l'hébergement de bases de données distantes, permettant un accès partagé entre le site web et l'application locale. Always Data offre un serveur FTP sur lequel je stocke tous mes scripts PHP car il n'a pas de limite. Possède une disponibilité de stockage de données en langage MySQL et offre une interface visuelle très sympas en utilisation phpMyAdmin. Grâce à Always Data tout mes données sont disponibles en ligne à n'importe quel moment.

MySQL : MySQL est un langage qui permet une création et gestion de bases de données et des tables efficace. Avant on utilisait le langage SQLite qui est le petit frère de MySQL est possède moins de fonctionnalité et n'est pas disponible sur Always Data.

FTP et Filezilla : FTP (File Transfert Protocol) permet de transférer rapidement les fichiers sur mon serveur site web vers le serveur distant en toute. Dans mon projet j'ai deux serveurs FTP, l'host LWS qui contient mes codes HTML les affiche sur mon site web et un deuxième host sur Always Data qui sauvegarde mes scripts PHP car il fallait un abonnement supérieur sur LWS pour mettre les scripts PHP avec les codes HTML. Filezilla permet la connexion sur les hosts FTP grâce à une adresse hôte, un nom d'utilisateur et un mot de passe pour que tout soit sécurisé.

Python : Python est un langage universel car il possède beaucoup de bibliothèque se qui augmente ces capacités, je l'utilise pour une gestion de tous mes capteurs connecter au Raspberry Pi 4. Je l'utilise aussi car je connaissais ses compétences liées aux grands nombres de bibliothèques.

Tkinter : Tkinter est une bibliothèque utilisée pour créer les interfaces graphiques sur la tablette du garde et de créer l'application d'employé.

mysql-connector-python : Cette bibliothèque permet aux applications Python de communiquer directement avec la base de données SQL.

bcrypt : bcrypt est un module qui permet de hacher les mots de passe avant de les stocker, garantissant pour la sécurité et la confidentialité des données clients.

pyinstaller : pyinstaller est utilisé pour convertir les scripts Python en exécutables en facilitant le déploiement sur les ordinateurs.

Un Raspberry Pi 4 4GB : C'est un petit ordinateur puissant qui possède des ports sur lequel on peut brancher des capteurs. C'est le cerveau de la maquette et de mon système de sécurité. Sa puissance de calcul permet de gérer le flux vidéo des caméras, la lecture des badges RFID et le traitement des données des capteurs gaz, eau et PIR, servos, Ses ports GPIO sont le lien physique entre le code et les composants électroniques. Tout cela sera contrôlé par une application de sécurité et on pourra se connecter à distance grâce à une connexion SSH.

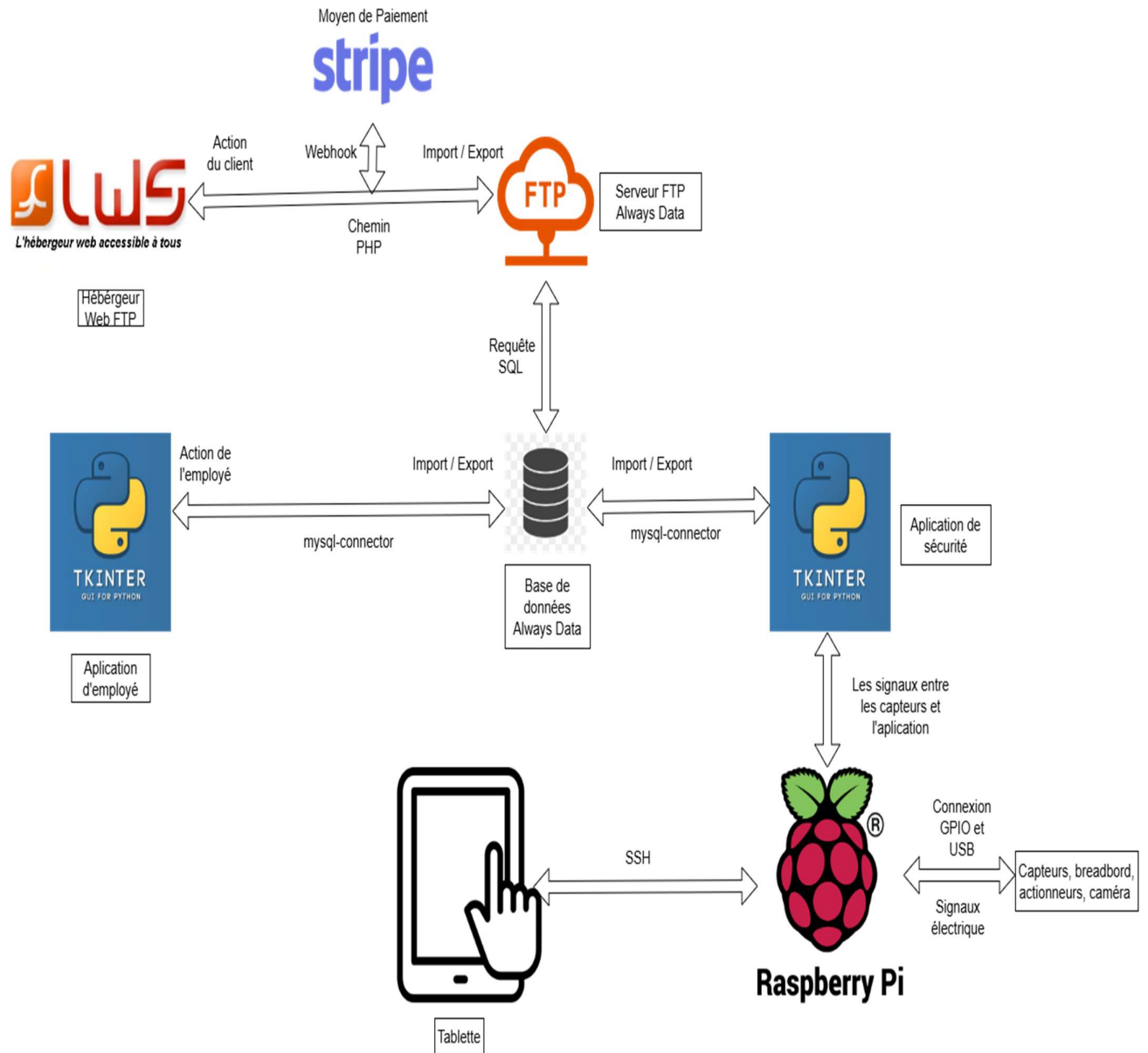
VNC Viewer : Une application qui permet de contrôler un ordinateur à distance via un réseau. Dans mon cas je l'utilise pour se connecter sur le Raspberry Pi 4 en SSH.

Open CV (Open Source Computer Vision) : La bibliothèque est spécialisée dans le traitement d'images et la vision par ordinateur en temps réel, je l'utilise pour la caméra qui permettra de voir la pièce du bâtiment.

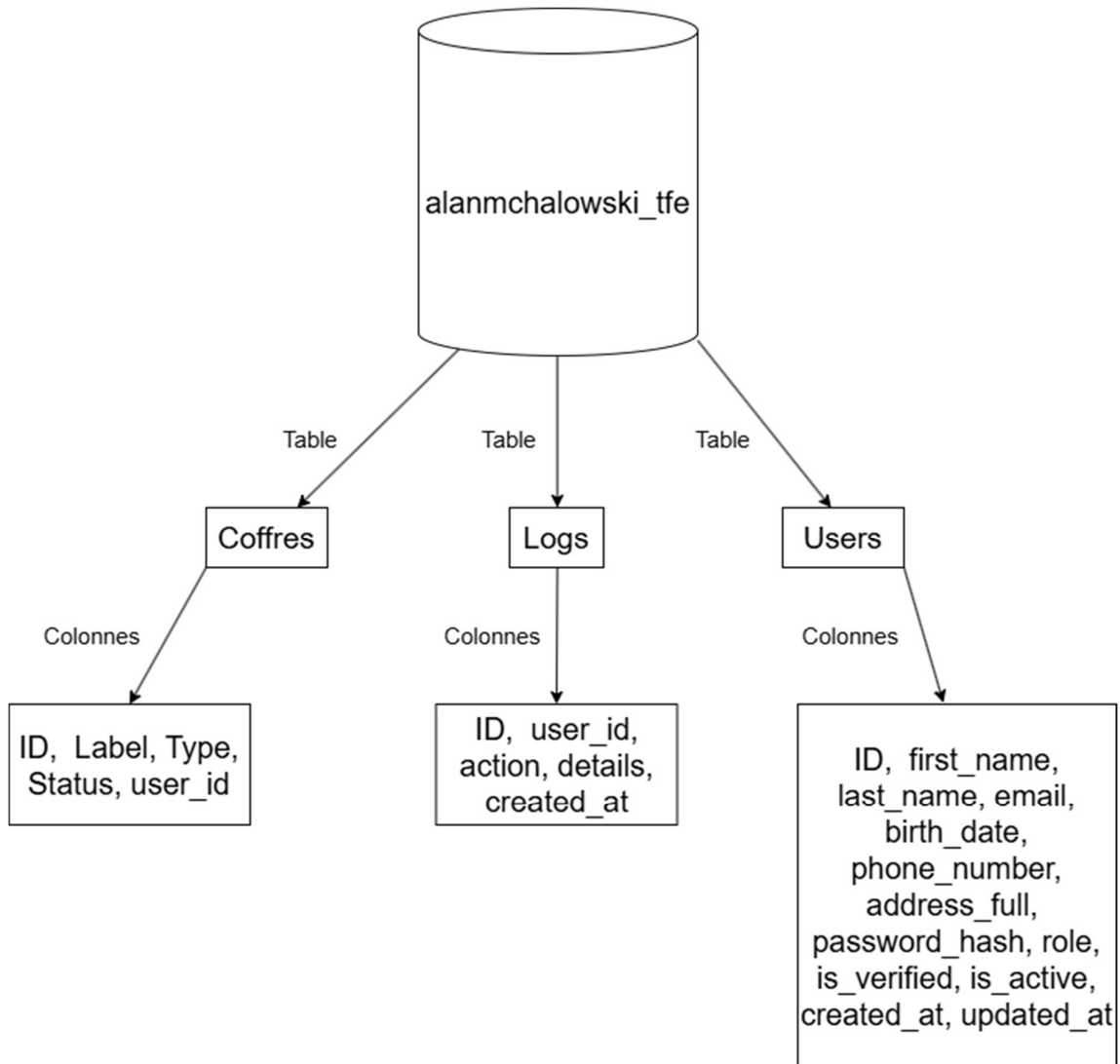
Threading : Outil qui exécute plusieurs parties d'un programme en même temps.

5. Conception du projet

5.1 Architecture globale



5.2 Modélisation de la base de données



5.3 Maquettes et Interfaces Utilisateurs

Avant de commencer la programmation de mes applications, je devais d'abord séparer les fonctionnements sur plusieurs pages car mettre tout sur une page ne serait efficace et beau.

Site Web :

- Page d'accueil : Informations sur l'entreprise, sur les offres et un menu navigable en haut
- Page de création de compte : L'endroit où le client créer un compte et met ses données personnelles.
- Page de connexion : Connexion à son compte

- Page Dashboard : Page personnel du client sur laquelle peut choisir un abonnement mensuel d'un coffret de stockages ou de livraisons, peut voir et modifier ses données, s'il prend un abonnement il voit son coffret, il peut annuler son abonnement coffret, Il peut se déconnecter.
- Page de paiement : Cela n'est pas modifiable car c'est une page de Stripe, email, données de la carte bancaire, nom et prénom
- Page de redirection : Une page rapide de redirection vers le Dashboard.

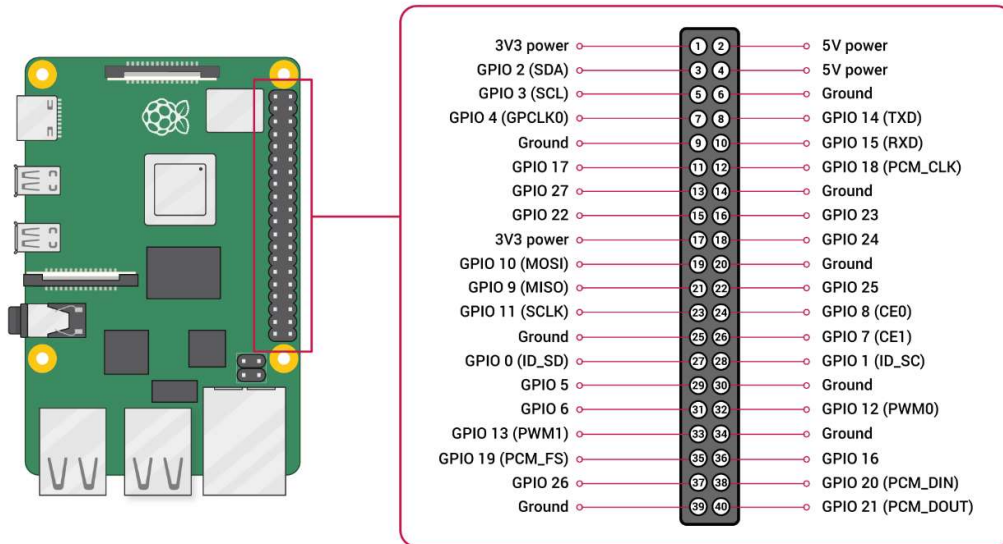
Application d'employé :

- Page d'accueil : 3 boutons de redirections s'affichent (Connexion, historique, Admin)
- Page de connexion : C'est une page de connexion pour les employeurs où ils mettent sont mail et mot de passe
- Page de réception : Endroit où l'employeur travaille, il voit tous les clients, peut modifier les données des clients, peut créer des clients, peut changer son propre mot de passe, peut voir tous les coffrets, peut faire une vérification d'un compte client, peut attribuer un coffret à un client et peut libérer un client.
- Page historique : C'est la page où on peut voir les logs.
- Page connexion admin : Connexion au compte admin
- Page création admin : Création d'un employé

Application de sécurité :

- Observation de la pièce du bâtiment avec la tablette grâce à la caméra.
- Ouverture de la porte de la pièce avec les coffrets de sûreté avec la tablette mais aussi avec un keypad si le garde de sécurité est absent.
- Ouverture de coffret personnel -> Confirmation sur la tablette + Badge du client + Keypad.
- Activation ou désactivation d'alarme de sécurité.
- Voir les états des portes, l'état des coffrets.

Branchement d'un Raspberry Pi 4

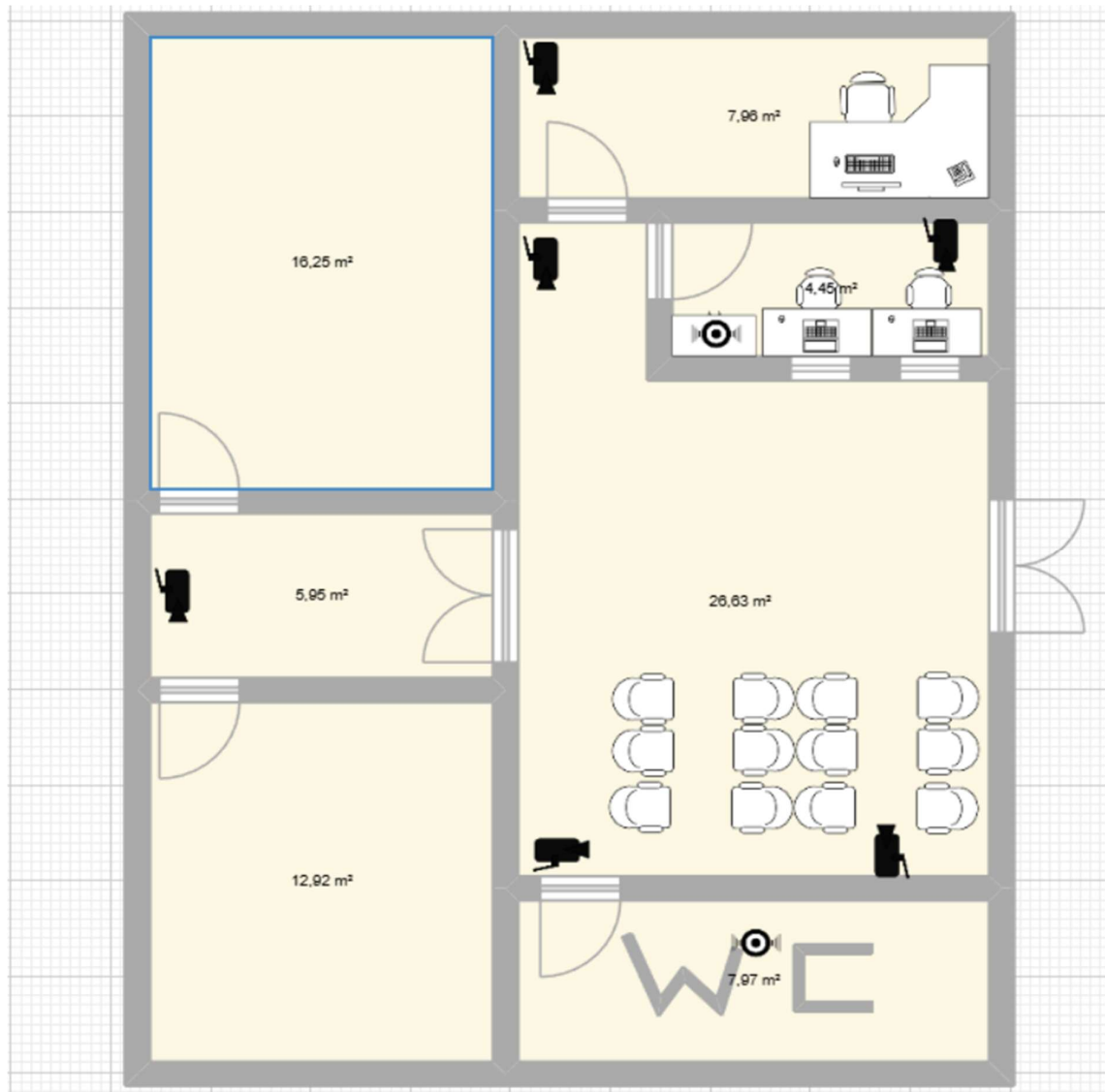


Composant	Broche VCC	Broche GND	Broche Signal / Data	Type d'interface
DHT 22	3.3V	GND	GPIO 4	One-Wire (Data)
Servo n°1	5V Externe	GND Commun	GPIO 12	PWM Matériel
Servo n°2	5V Externe	GND Commun	GPIO 13	PWM Matériel
PIR n°1	5V	GND	GPIO 17	Digital Input
PIR n°2	5V	GND	GPIO 27	Digital Input

Composant	Broche VCC	Broche GND	Broche Signal / Data	Type d'interface
Capteur Gaz	5V	GND	GPIO 22	Digital (DO)
Vibration	3.3V	GND	GPIO 5	Digital Input
Caméra USB	Port USB	N/A	N/A	USB

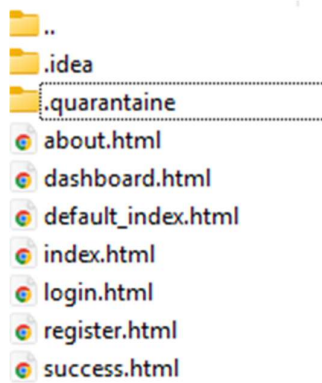
Composant	Interface	Broches spécifiques
RFID (RC522)	SPI	SDA: 8 , SCK: 11 , MOSI: 10 , MISO: 9 , RST: 25
Keypad (4x4)	GPIO	Lignes: 6, 16, 26, 18 / Colonnes: 20, 21, 14, 15

Plan de la maquette (batiment)

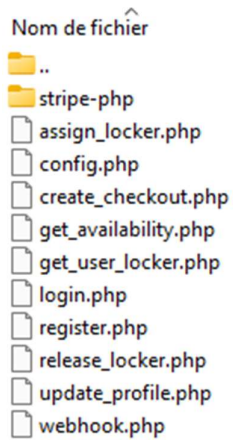


5.4 Organisation des fichiers et structure du code

Serveur FTP de LWS :



Serveur Always Data : Dans le dossier www



6. Réalisation (développement)

6.1 Explication du fonctionnement

En pratique, le système fonctionne comme une boucle continue d'échanges d'informations sécurisées entre le site, les applications et la base de données. Lorsqu'un client s'inscrit et paie son abonnement via le site web, le paiement doit être accepté par Stripe avec une clé privée avec un id unique, ensuite le script PHP enregistre ses données (avec un mot de passe haché) dans la base de données MySQL hébergée sur Always Data. Le client à son arriver à la réception passe par une vérification de donner grâce à l'application d'employé. Du côté du bâtiment physique, le Raspberry Pi exécute un script Python principal en arrière-plan. Ce script interroge la base de données pour synchroniser les droits d'accès. Lorsqu'un client se présente et badge sa carte RFID, le Raspberry Pi lit l'identifiant, le compare avec la base de données et attend la validation via le code PIN. Si les données sont correctes, une notification apparaît sur l'application Tkinter de la tablette du garde. Ce n'est qu'après la validation finale du garde que le script Python envoie un signal au port GPIO pour actionner le servomoteur et ouvrir la porte du coffret. Si le garde de sécurité est absent, il n'y a pas de validation sur la tablette. Les capteurs environnementaux scannent les pièces en continu.

6.2 Présentation des fonctionnalités implémentées

Le site Web Client & Système de Paiement constitue le premier point de contact avec l'utilisateur pour la gestion de son abonnement à distance. La création et gestion de compte est développée en HTML et CSS pour l'interface visuelle. Le langage JavaScript vérifie si l'ensemble des données saisies côté client (présence du caractère @ pour l'adresse mail, exigences de sécurité de 8 caractères minimum comprenant une majuscule et un chiffre pour le mot de passe). Inscription des clients avec vérification de la robustesse du mot de passe et hachage via Bcrypt. Les scripts PHP, hébergés sur le serveur FTP de Always Data, sécurisent la session et font le pont avec la base de données pour enregistrer ou modifier les informations du profil client. L'intégration de l'API Stripe gère l'écosystème des transactions (abonnements mensuels ou paiements uniques) en toute légalité en Belgique. Grâce au mécanisme des clé ID Webhooks, Stripe envoie une notification sécurisée en tâche de fond au script PHP lors de la réussite d'un paiement, déclenchant de manière automatisée l'attribution d'un numéro de coffret unique au client dans le système.

J'ai créé une base de données chez Always Data, la base de données fonctionne sous le système de gestion MySQL. Cela permet un accès partagé et simultané, le site web distant (PHP) et

l'application locale Python sur Raspberry Pi lisent et écrivent dans les mêmes tables en temps réel. Elle stocke de manière les informations des clients, des employés, de l'administrateur, garde de sécurité, l'état physique des coffrets, grâce à un système de logs qui permet de voir l'historique en cas d'erreurs ou de vol.

L'application d'employé est une interface de réception de client sur place pour les personnes âgées en difficulté avec les outils en ligne, l'employé peut créer manuellement le compte d'un client depuis cette interface en lui attribuant un mot de passe temporaire. L'application permet à l'employeur de vérifier le statut des paiements des clients venant de s'inscrire en ligne. Après vérification d'identité, l'employé utilise l'interface pour associer physiquement un badge RFID au compte du client. Pour le développement de cette application j'ai utilisé le langage Python avec la bibliothèque graphique Tkinter pour l'interface de gestion d'utilisateur et de coffret. Elle utilise mysql-connector-python pour modifier directement la base de données Always Data et le module bcrypt pour hacher et sécuriser les mots de passe des clients créés par les employés. Pour la création d'un employé, il se trouve une interface administrateur.

Le système de sécurité et la maquette représentent la sécurité physique du bâtiment et le contrôle d'accès en temps réel. Le Raspberry Pi 4 contrôle l'accès à la pièce sécurisée. Si le garde est absent, le client accède à la salle en saisissant son code PIN sur un Keypad. Le script Python valide le PIN auprès de la base de données. Si le garde est présent, il peut superviser les arrivées en direct via le flux des caméras et l'affichage des logs, puis ouvrir la porte à distance. Une fois à l'intérieur de la salle, l'accès au coffret individuel exige une validation conjointe rigoureuse : la lecture du badge RFID du client, la saisie de son code PIN à 4 chiffres pour éviter la suffisance du le vol de badge, et l'autorisation finale d'un garde de sécurité via l'interface tactile Tkinter de sa tablette de sécurité. Lorsqu'un livreur se présente, il donne les noms des destinataires. L'employé lui donne une liste de numéros de coffrets correspondants. Dès que le livreur dépose le colis et referme le coffret grâce à un capteur de mouvement, le système envoie automatiquement une notification au client pour l'avertir que son colis est disponible. Les ports GPIO du Raspberry Pi scannent en arrière-plan et en continu les capteurs de gaz, d'inondation (eau) et de mouvement (PIR). En cas d'anomalie, le système active les servomoteurs de sécurité, déclenche l'alarme et commence une alerte d'évacuation. L'accès distant via SSH utiliser sur VNC Viewer permet la maintenance de la sécurité. La distribution finale des scripts Python est simplifiée sous forme d'exécutables grâce à pyinstaller.

6.3 Extraits de code commentés

register.html

```
637 <script>
638     function showAlert(message, type) {
639         const alertContainer = document.getElementById('alertContainer');
640         const alertClass = type === 'error' ? 'alert-error' : 'alert-success';
641         alertContainer.innerHTML = `<div class="alert ${alertClass}">${message}</div>`;
642     }
643
644     function validatePassword(password) { //pour valider le mot de passe
645         const regex = /^(?=.*[A-Z])(?=.*\d){8,}$/;
646         return regex.test(password);
647     }
648
649     document.getElementById('registerForm').addEventListener('submit', function(e) {
650         e.preventDefault(); // Empêche la page de recharger
651
652         const password = document.getElementById('password').value;
653         const confirmPassword = document.getElementById('confirmPassword').value;
654
655         // 1. Validations du mot de passe
656         if (!validatePassword(password)) {
657             showAlert('Le mot de passe doit contenir au moins 8 caractères, une majuscule et un chiffre.', 'error');
658             return;
659         }
660
661         if (password !== confirmPassword) {
662             showAlert('Les mots de passe ne correspondent pas.', 'error');
663             return;
664         }
665
666         // 2. On récupère automatiquement tous les champs du formulaire
667         const formData = new FormData(this);
668         ...
669
670         // 3. On envoie les données au serveur AlwaysData
671         fetch('https://alanmichalowski.alwaysdata.net/register.php', {
672             method: 'POST',
673             body: formData
674         })
675         .then(response => response.json()) // On lit la réponse (Succès ou Erreur)
676         .then(data => {
677             if (data.status === 'success') {
678                 showAlert(data.message + ' Redirection vers la connexion...', 'success');
679                 setTimeout(() => {
680                     window.location.href = 'login.html';
681                 }, 1500);
682             } else {
683                 // Si le PHP renvoie une erreur (ex: email déjà pris)
684                 showAlert(data.message, 'error');
685             }
686         })
687         .catch(error => {
688             console.error('Erreur:', error);
689             showAlert('Impossible de joindre le serveur AlwaysData.', 'error');
690         });
691
692         // Vérifier si déjà connecté
693         if (localStorage.getItem('isLoggedIn') === 'true') {
694             window.location.href = 'dashboard.html';
695         }
696     </script>
```

success.html

```
58 <script>
59 // 1. On donne 4 secondes au serveur pour traiter le Webhook Stripe
60 setTimeout(() => {
61     const userData = JSON.parse(localStorage.getItem('userData'));
62
63     if (userData && userData.id) {
64         // 2. On met à jour les données locales AVANT de retourner au dashboard
65         fetch(`https://alanmichalowski.alwaysdata.net/get_user.php?id=${userData.id}`)
66             .then(res => res.json())
67             .then(data => {
68                 if (data.status === 'success') {
69                     // On écrase les anciennes données par les nouvelles (avec le coffret !)
70                     localStorage.setItem('userData', JSON.stringify(data.user));
71                 }
72                 // 3. Redirection vers le dashboard
73                 window.location.href = 'dashboard.html';
74             })
75             .catch(() => {
76                 // En cas d'erreur réseau, on redirige quand même
77                 window.location.href = 'dashboard.html';
78             });
79     } else {
80         window.location.href = 'dashboard.html';
81     }
82 }, 4000); // 4000 millisecondes = 4 secondes
83 </script>
```

Login.html

```
274 function showAlert(message, type) {
275     const alertContainer = document.getElementById('alertContainer');
276     const alertClass = type === 'error' ? 'alert-error' : 'alert-success';
277
278     alertContainer.innerHTML = `<div class="alert ${alertClass}">${message}</div>`;
279 }
280
281 // Interception de l'événement "submit" du formulaire de connexion
282 document.getElementById('loginForm').addEventListener('submit', function(e) {
283     e.preventDefault();
284
285     const formData = new FormData(this); // Prépare les données du formulaire (email, password)
286
287     // Envoi d'une requête HTTP POST vers le serveur PHP sur AlwaysData
288     fetch('https://alanmichalowski.alwaysdata.net/login.php', {
289         method: 'POST',
290         body: formData
291     })
292     .then(response => response.json()) // Conversion de la réponse serveur en objet JSON
293     .then(data => {
294         if (data.status === 'success') {
295             // GESTION DE LA SESSION CLIENT
296             // Stockage permanent dans le navigateur pour identifier l'utilisateur sur les autres pages
297             localStorage.setItem('isLoggedIn', 'true');
298             localStorage.setItem('userData', JSON.stringify(data.user)); // Sérialisation des infos client
299
300             showAlert('Connexion réussie ! Redirection...', 'success');
301
302             // Redirection différée pour laisser le temps à l'utilisateur de lire le message de succès
303             setTimeout(() => {
304                 window.location.href = 'dashboard.html';
305             }, 1000);
306         } else {
307             // Affichage de l'erreur renvoyée par le script PHP (ex: "Identifiants incorrects")
308             showAlert(data.message, 'error');
309         }
310     })
311     .catch(error => {
312         // Sécurité : gestion des pannes réseau ou serveur injoignable
313         console.error('Erreur:', error);
314         showAlert('Erreur de connexion au serveur.', 'error');
315     });
316 });
```

dashboard.html

```
200 // AFFICHAGE DU COFFRET (La clé du pr)
201 const label = userData.locker_label;
202 const statusDisplay = document.getElementById('coffretStatus');
203 const cancelBtn = document.getElementById('cancelLockerBtn');
204 // Test : si le label existe et n'est pas vide
205 if (label && label !== "null" && label !== "") {
206     statusDisplay.textContent = label; // Affiche le nom du coffret
207     statusDisplay.style.color = "var(--accent-color)"; // Met en vert (succès)
208     cancelBtn.style.display = 'inline-block'; // Affiche le bouton "Libérer"
209 } else {
210     statusDisplay.textContent = "Aucun"; // État par défaut
211     statusDisplay.style.color = "white";
212     cancelBtn.style.display = 'none'; // État par défaut
213 }
214

216 // 3. Rafraîchissement depuis la BDD
217 function refreshUserProfile() {
218     const userData = JSON.parse(localStorage.getItem('userData'));
219     if (!userData || !userData.id) return;
220     // Requête vers le script PHP pour récupérer les dernières infos du client
221     fetch('https://alanmichalowski.alwaysdata.net/get_user.php?id=${userData.id}')
222         .then(res => res.json())
223         .then(data => {
224             if (data.status === 'success') { // Mise à jour du stockage local avec les nouvelles données (ex: après un paiement)
225                 localStorage.setItem('userData', JSON.stringify(data.user));
226                 loadUserData(); // Relance l'affichage des informations à l'écran
227                 console.log("Données synchronisées !");
228             }
229         })
230         .catch(err => console.error("Erreur sync:", err));
```

```

233 // 4. Récupération des stocks |
234 function updateAvailability() {
235     fetch('https://alanmichalowski.alwaysdata.net/get_availability.php')
236         .then(res => res.json())
237         .then(data => {
238             ['petit', 'moyen', 'grand'].forEach(type => {
239                 const count = data[type] || 0;
240                 const stockEl = document.getElementById(`stock-${type}`);
241                 const btnEl = document.querySelector(`#offer-${type} .btn`);
242                 stockEl.textContent = `${count} coffrets disponibles`;
243                 if (count <= 0) {
244                     stockEl.style.color = 'red';
245                     btnEl.disabled = true;
246                     btnEl.style.background = "#ccc";
247                 }
248             });
249         });
250 }

```

```

252 // 5. Redirection Stripe
253 function assignLocker(type) {
254     const userData = JSON.parse(localStorage.getItem('userData'));
255     if (userData.locker_label) {
256         showAlert("Vous avez déjà un coffret : " + userData.locker_label, "error");
257         return;
258     }
259
260     const formData = new FormData();
261     formData.append('user_id', userData.id);
262     formData.append('type', type);
263
264     fetch('https://alanmichalowski.alwaysdata.net/create_checkout.php', {
265         method: 'POST',
266         body: formData
267     })
268     .then(res => res.json())
269     .then(data => {
270         if (data.url) window.location.href = data.url;
271         else showAlert("Erreur paiement", "error");
272     });
273 }
274

```

```

275 // 6. Libérer coffret
276 function releaseLocker() {
277     if (!confirm("Voulez-vous libérer votre coffret ?")) return;
278     const userData = JSON.parse(localStorage.getItem('userData'));
279     const formData = new FormData();
280     formData.append('user_id', userData.id);
281
282     fetch('https://alanmichalowski.alwaysdata.net/release_locker.php', {
283         method: 'POST',
284         body: formData
285     })
286     .then(res => res.json())
287     .then(data => {
288         if (data.status === 'success') {
289             refreshUserProfile(); // On recharge les infos
290             showAlert("Coffret libéré !", "success");
291         }
292     });
293 }

```

config.php

```
1 <?php
2 $host = 'mysql-alanmichalowski.alwaysdata.net';
3 $db = 'alanmichalowski_tfe';
4 $user = 'alanmichalowski';
5 $pass = 'PolskaGurom1234';
6 $charset = 'utf8mb4';
7
8 $dsn = "mysql:host=$host;dbname=$db;charset=$charset";
9
10 try { // Création de l'objet PDO pour communiquer avec MySQL sur AlwaysData
11     $pdo = new PDO($dsn, $user, $pass);
12     // Activation du mode exception pour détecter immédiatement toute erreur SQL
13     $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
14 } catch (\PDOException $e) {
15     // On renvoie l'erreur au format JSON pour que LWS la comprenne
16     die(json_encode(["status" => "error", "message" => "Erreur de connexion à la base de données."]));
17 }
18 ?>
```

register.php

```
<?php
// 1. Autoriser ton site LWS à envoyer des infos ici
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Methods: POST");
header("Content-Type: application/json; charset=UTF-8");

require 'config.php';

// 2. Vérifier qu'on reçoit bien des données
if ($_SERVER['REQUEST_METHOD'] === 'POST') {

    // 3. Récupérer les vraies données du formulaire HTML
    $firstName = $_POST['prenom'] ?? '';
    $lastName = $_POST['nom'] ?? '';
    $email = $_POST['email'] ?? '';
    $birth = $_POST['dateNaissance'] ?? '';
    $phone = $_POST['gsm'] ?? '';
    $address = $_POST['adresse'] ?? '';
    $passRaw = $_POST['password'] ?? '';

    // Hachage du mot de passe
    $passwordHash = password_hash($passRaw, PASSWORD_DEFAULT);

    // Requête SQL
    $sql = "INSERT INTO users (first_name, last_name, email, birth_date, phone_number, address_full, password_hash)
        | VALUES (:fname, :lname, :email, :birth, :phone, :address, :pass)";

    try {
        $stmt = $pdo->prepare($sql);
        $stmt->execute([
            ':fname' => $firstName,
            ':lname' => $lastName,
            ':email' => $email,
            ':birth' => $birth,
            ':phone' => $phone,
            ':address' => $address,
            ':pass' => $passwordHash
        ]);

        // --- ENVOI MAIL BIENVENUE ---
        $to = $email;
        $subject = "Bienvenue chez SafeBox Pro !";
        $message = "
```

```

// JOINTURE : On récupère l'user + le label du coffre s'il en a un
$sql = "SELECT u.*, c.label AS locker_label
        FROM users u
        LEFT JOIN coffres c ON u.id = c.user_id
        WHERE u.email = :email";

try {
    $stmt = $pdo->prepare($sql);
    $stmt->execute([':email' => $email]);
    $user = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($user && password_verify($password, $user['password_hash'])) { // Sécurité
        unset($user['password_hash']); // On supprime le hash avant d'envoyer au JS par sécurité

        echo json_encode([
            "status" => "success",
            "message" => "Connexion réussie !",
            "user" => $user
        ]);
    } else {
        echo json_encode(["status" => "error", "message" => "Identifiants incorrects."]);
    }
} catch (PDOException $e) {
    echo json_encode(["status" => "error", "message" => "Erreur SQL : " . $e->getMessage()]);
}

```

webhook.php

```

5 $stripeSecretKey = 'sk_test_51Tbr1sI68c5vUoxBbCyfX3XtC6v6uufdNg1rnASaR2LvUB19MUQhgTZ3JryOd5ebh2rvsvf8Ztsi1cF8ewaTq200HRSNkcaJ';
6 $endpoint_secret = 'whsec_GB3PSvv05LmsEHjnIqDUnbxjcr1BUr5G';
7
8 \Stripe\Stripe::setApiKey($stripeSecretKey);
9
0 $payload = @file_get_contents('php://input');
1 $sig_header = $_SERVER['HTTP_STRIPE_SIGNATURE'] ?? '';

if ($event->type == 'checkout.session.completed') {
    $session = $event->data->object;

    $user_id = $session->client_reference_id;
    $type = $session->metadata['coffret_type'] ?? 'petit';

    try {
        // 1. Recherche d'un coffre libre
        // On utilise une requête préparée pour sélectionner un coffre spécifique.
        // "type = ?" permet de filtrer (petit, moyen, grand) selon l'achat du client.
        // "status = 'libre'" garantit qu'on ne propose pas un coffre déjà pris.
        // "LIMIT 1" est crucial : on ne veut en récupérer qu'un seul à la fois.
        $stmt = $pdo->prepare("SELECT id, label FROM coffres WHERE type = ? AND status = 'libre' LIMIT 1");
        $stmt->execute([$type]);
        $locker = $stmt->fetch(); // On stocke les infos du coffre trouvé dans la variable $locker

        // Vérification de sécurité : on ne procède que si un coffre a été trouvé
        // ET que l'identifiant de l'utilisateur est bien présent.
        if ($locker && !empty($user_id)) {

            // 2. Attribution du coffre
            // On met à jour la table 'coffres' pour marquer le coffre comme 'occupé'.
            // On lie l'ID de l'utilisateur (user_id) à la ligne du coffre (WHERE id = ?).
            $update = $pdo->prepare("UPDATE coffres SET status = 'occupé', user_id = ? WHERE id = ?");

            // Exécution de la mise à jour avec les données sécurisées
            $update->execute([$user_id, $locker['id']]);
        }
    }
}

```

get_availability.php

```
try {
    // On compte combien de coffres sont 'libre' pour chaque type
    $sql = "SELECT type, COUNT(*) as dispo FROM coffres WHERE status = 'libre' GROUP BY type";
    $stmt = $pdo->query($sql);
    $results = $stmt->fetchAll(PDO::FETCH_ASSOC);

    // On prépare une réponse propre
    $availability = [
        'petit' => 0,
        'moyen' => 0,
        'grand' => 0
    ];

    foreach ($results as $row) {
        $availability[$row['type']] = (int)$row['dispo'];
    }

    echo json_encode($availability);

} catch (PDOException $e) {
    echo json_encode(["error" => $e->getMessage()]);
}
?>
```

Create_checkout.php

```
// Clé secrète
\Stripe\Stripe::setApiKey('sk_test_51TbrlsI68c5vUoxBbCyfX3XtC6v6uufdNg1rnAsaR2LvUB19MUQhgTZ3JryOd5ebh2rvsvfn8Ztsi1cF8ewaTq200HR5NkcaJ');

$YOUR_DOMAIN = 'https://michalowski.kialan.be';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // On récupère les infos envoyées par le JavaScript (fetch)
    $user_id = $_POST['user_id'] ?? '';
    $type = $_POST['type'] ?? '';

    // SÉCURITÉ : Si l'ID utilisateur est vide, on s'arrête tout de suite
    if (empty($user_id)) {
        echo json_encode(['error' => 'Erreur : ID utilisateur manquant. Connectez-vous à nouveau.']);
        exit;
    }

    $prices = [
        'petit' => 'price_1TbrtvI68c5vUoxBA1xTBvbZ',
        'moyen' => 'price_1TbrvKI68c5vUoxB649kDMFC',
        'grand' => 'price_1Tbry6I68c5vUoxBSaZDu5B9'
    ];

    if (!isset($prices[$type])) {
        echo json_encode(['error' => 'Type de coffret invalide.']);
        exit;
    }

    try {
        $checkout_session = \Stripe\Checkout\Session::create([
            'line_items' => [[
                'price' => $prices[$type],
                'quantity' => 1,
            ]],
            'mode' => 'subscription',
            // C'est ici que le lien se fait avec ton Webhook :
            'client_reference_id' => (string)$user_id,
            'metadata' => [
                'coffret_type' => $type
            ],
            'success_url' => $YOUR_DOMAIN . '/success.html',
            'cancel_url' => $YOUR_DOMAIN . '/dashboard.html?payment=cancel',
        ]);

        echo json_encode(['url' => $checkout_session->url]);
    }
}
```

PageConnexion.py

```
57 # --- LOGIQUE DE CONNEXION RÉPARÉE POUR BCRIPT ($2y$) ---
58 def soumettre():
59     email = entry_email.get().strip()
60     mdp_saisi = entry_mdp.get()
61
62     if not email or not mdp_saisi:
63         messagebox.showwarning("Attention", "Veuillez remplir tous les champs.")
64         return
65
66     try:
67         conn = pymysql.connect(
68             host='mysql-alanmichalowski.alwaysdata.net',
69             db='alanmichalowski_tfe',
70             user='alanmichalowski',
71             password='PolskaGurom1234',
72             charset='utf8mb4'
73         )
74         cursor = conn.cursor()
75
76         # 1. On récupère les infos ET le hash stocké pour cet email
77         query = "SELECT id, first_name, last_name, password_hash FROM users WHERE email=%s"
78         cursor.execute(query, (email,))
79         utilisateur = cursor.fetchone()
80
81         if utilisateur:
82             user_id, prenom, nom, hash_stocke = utilisateur
83
84             # 2. Vérification du mot de passe avec Bcrypt
85             # password_verify($mdp_saisi, $hash_stocke) en PHP
86             if bcrypt.checkpw(mdp_saisi.encode('utf-8'), hash_stocke.encode('utf-8')):
87                 nom_complet = f"{prenom} {nom}"
88
89                 # Log de connexion
90                 cursor.execute("INSERT INTO logs (user_id, action, details) VALUES (%s, %s, %s)",
91                               (user_id, "Connexion", f"L'employé {nom_complet} s'est connecté."))
92                 conn.commit()
93
94                 messagebox.showinfo("Succès", f"Bienvenue {nom_complet} !")
95                 ouvrir_page_reception(user_id, nom_complet)
96             else:
97                 messagebox.showerror("Erreur", "Mot de passe incorrect.")
98         else:
99             messagebox.showerror("Erreur", "Email inconnu.")
100
101     cursor.close()
102     conn.close()
103
```

PageconnexionAdmin.py

```
64 # 1. On cherche l'utilisateur ADMIN par son email uniquement et son role
65 query = "SELECT id, first_name, password_hash FROM users WHERE email=%s AND role='admin'"
66 cursor.execute(query, (email,))
67 admin = cursor.fetchone()
```

PageReception.py

```
L40 # --- Pop-up : NOUVEAU CLIENT ---
L41 def pop_up_nouveau_client():
L42     """Ouvre une fenêtre modale pour créer un nouveau compte client."""
L43     # Création de la fenêtre secondaire (modale)
L44     pop = tk.Toplevel(pr)
L45     pop.title("Nouveau Client")
L46     pop.geometry("500x650")
L47     pop.configure(bg="white")
L48
L49     # Titre affiché en haut de la fenêtre
L50     tk.Label(pop, text="Créer un compte Client", font=("Arial", 16, "bold"), bg="white").pack(pa
L51
L52     champs = {} # Dictionnaire pour stocker les widgets Entry par clé
L53
L54     # Liste des champs à afficher : (label visible, clé interne)
L55     liste_labels = [
L56         ("Nom", "last"), ("Prénom", "first"), ("Email", "email"),
L57         ("Téléphone", "phone"), ("Mot de passe", "pwd"),
L58         ("Naissance (AAAA-MM-JJ)", "birth"), ("Adresse", "addr")
L59     ]
L60
L61     for txt, key in liste_labels:
L62         tk.Label(pop, text=txt + " :", bg="white").pack(anchor="w", padx=50)
L63         # Masquage des caractères pour le champ mot de passe uniquement
L64         show_c = "." if key == "pwd" else ""
L65         ent = tk.Entry(pop, width=40, font=("Arial", 11), bg="#F9F9F9", show=show_c)
L66         ent.pack(pady=5)
L67         champs[key] = ent # Stockage du widget dans le dictionnaire
L68
L69     def sauvegarder():
L70         """Insère le nouveau client en base avec son mot de passe hashé."""
L71         try:
L72             conn = db_connect()
L73             cursor = conn.cursor()
L74
L75             # Hachage sécurisé du mot de passe avec bcrypt avant stockage
L76             mdp_client_hashe = hasher_bcrypt(champs["pwd"].get())
L77
L78             # Requête d'insertion avec tous les champs du formulaire
L79             query = """
L80             INSERT INTO users (last_name, first_name, email, phone_number, password_hash,
L81             birth_date, address_full, role, is_verified, is_active,
L82             created_at, updated_at)
L83             VALUES (%s, %s, %s, %s, %s, %s, %s, %s, 'client', 1, 1, NOW(), NOW())
L84             """
L85             cursor.execute(query, (
L86                 champs["last"].get(), champs["first"].get(), champs["email"].get(),
L87                 champs["phone"].get(), mdp_client_hashe,
```

```

301 # Création du frame pour l'onglet Coffrets
302 tab_coffres = tk.Frame(onglets, bg="white")
303 onglets.add(tab_coffres, text=" Coffrets")
304
305 # Tableau des coffrets avec 5 colonnes
306 tree_cof = ttk.Treeview(tab_coffres, columns=("id", "label", "type", "statut", "client"), show="headings")
307 for col, tit in zip(("id", "label", "type", "statut", "client"), ("ID", "Numéro", "Type", "Statut", "Client")):
308     tree_cof.heading(col, text=tit)
309 tree_cof.pack(fill="both", expand=True, padx=10, pady=10)
310
311
312 def charger_coffres():
313     """Vide puis recharge le tableau des coffrets depuis la base de données."""
314     for item in tree_cof.get_children():
315         tree_cof.delete(item)
316     try:
317         conn = db_connect()
318         cursor = conn.cursor()
319         # Jointure avec users pour afficher le nom du client locataire
320         # CASE pour convertir le statut numérique en texte lisible
321         query = """
322             SELECT c.id, c.label, c.type,
323                    CASE WHEN c.status = 0 THEN ' Disponible' ELSE ' Loué' END,
324                    IFNULL(CONCAT(u.first_name, ' ', u.last_name), '---'),
325                    c.status
326             FROM coffres c
327             LEFT JOIN users u ON c.user_id = u.id
328         """
329         cursor.execute(query)
330         for row in cursor.fetchall():
331             # Insertion des 5 premières colonnes ; le statut brut (row[5]) sert de tag
332             # pour pouvoir le relire sans parser le texte affiché
333             tree_cof.insert("", "end", values=row[:5], tags=(str(row[5]),))
334         conn.close()
335     except Exception as e:
336         print(e)
337
338
339 def pop_up_attribuer():
340     """Ouvre une fenêtre modale pour attribuer un coffret disponible à un client vérifié."""
341     selection = tree_cof.selection()
342     if not selection:
343         return messagebox.showwarning("Erreur", "Sélectionnez un coffret")
344
345     # Récupération de l'ID et du numéro du coffret sélectionné
346     c_id, c_label = tree_cof.item(selection[0])['values'][:2]
347     # Le statut brut est stocké dans les tags (0 = disponible, 1 = loué)
348     c_status_brut = tree_cof.item(selection[0], 'tags')[0]
349
350     # Bloquer l'attribution si le coffret est déjà loué
351     if int(c_status_brut) != 0:
352         return messagebox.showerror("Erreur", "Déjà loué.")
353
354     pop = tk.Toplevel(pr)
355     pop.geometry("400x300")
356     tk.Label(pop, text=f"Coffre n°{c_label}", font=("Arial", 12, "bold")).pack(pady=10)
357
358     # Liste déroulante pour choisir le client
359     combo = ttk.Combobox(pop, width=30, state="readonly")
360     combo.pack(pady=5)
361
362     clients_map = {} # Dictionnaire : texte affiché → id du client
363
364     try:
365         conn = db_connect()
366         cursor = conn.cursor()
367         # Seuls les clients vérifiés peuvent recevoir un coffret
368         cursor.execute("SELECT id, first_name, last_name FROM users WHERE role='client' AND is_verified=1")
369         for cid, f, l in cursor.fetchall():
370             txt = f"{f} {l} (ID:{cid})"
371             clients_map[txt] = cid # Association texte → id pour récupérer l'id au moment de la sélection
372         combo['values'] = list(clients_map.keys())
373         conn.close()
374     except:
375         pass
376
377 def valider():
378     """Enregistre l'attribution du coffret au client sélectionné."""
379     if not combo.get():
380         return # Aucun client sélectionné : on ne fait rien
381
382     client_id_choisi = clients_map[combo.get()] # Récupération de l'id via le dictionnaire
383     try:
384         conn = db_connect()

```



```

def update_sensors_thread(self):
    def loop():
        while True:
            # Lecture réelle des GPIO
            pir_state = GPIO.input(PIN_PIR)
            vib_state = GPIO.input(PIN_VIBRATION)

            if self.is_armed and (pir_state or vib_state):
                self.alarm_active = True

            self.sensor_data['temp']
            self.sensor_data['gas']
            self.sensor_data['dist']

            time.sleep(2)
    threading.Thread(target=loop, daemon=True).start()

```

Commande que je devais réaliser sur les applications

- pip install mysql-connector-python
- pip install bcrypt
- pip install pyinstaller
- pyinstaller --noconsole --onefile --name "MonApplication" main.py
- python -m PyInstaller --noconsole --onefile --name "Secure Boxe Employeur" PageAccueil.py
- pip install opencv-python pillow adafruit-circuitpython-dht spidev rpi.gpio

6.4 Difficultés rencontrées

- Mauvaise interprétation de Visual Studio Code quand je mélange plusieurs langages.
- Manque de possibilité de mettre les fichiers PHP, sur le FTP de LWS car il fallait payer un abonnement de plus.
- Mélange entre les deux clé Stripe, la clé secrète et la clé endpoint alors que les fonctionnements sont différents, la clé secrète permet au serveur Always Data d'envoyer des ordres à Stripe comme la création de la session des paiements. La clé endpoint est la clé de signature (confirmation) que le paiement à était effectuer. Sur le site web de Stripe, les deux sont nommé comme clé secrète, cela m'a induit en erreurs et à provoquer des sessions de paiements rejeter alors que l'interface de paiements ne s'affichait pas.

- Plusieurs versions de Python installer sur mon ordinateur à causer des bugs, l'activation de la mauvaise version et l'installation multiples de la même bibliothèque.
- Une alimentation de Raspberry Pi 4 insuffisante pour les branchements.
- Utilisation de deux méthodes d'hachage différent, cela à diviser mon projet en deux car il y avait des périphériques qui pouvait se connecter et l'autres partie ne pouvait plus.

6.5 Solutions apportées

- Séparer les codes dans des dossier différent pour éviter une mauvaise interprétation de Visual Studio Code quand je mélange plusieurs langages.
- Migration du backend PHP vers Always Data car celui-ci ne possède pas de limitation comme LWS.
- Une compréhension de fonctionnement de clés Stripe.
- La désinstallation de tout les versions de python, ensuite une réinstallation d'une.
- Achat d'un bloc d'alimentation externe 5 volt qui permet de connecter beaucoup de capteurs.
- Utilisation que du hachage bcrypte qui était meilleur entre les deux.

7. Tests, Débogage et Validations

7.1 Protocoles de tests réalisés.

Tests de l'interface Web et de la Base de données, J'ai saisi de données erronées ou incomplètes dans le formulaire d'inscription, création de plusieurs utilisateurs sur un mail. Vérifier le hachage de mot de passe avec bcrypt

Tests de la passerelle de paiement (Stripe). Essaie de faire beuguer les paiements, saisir des cartes expirées ou payer avec une carte qui possède des fonds insuffisants.

Tests de l'application d'employé, tentatives de connexion avec des identifiants erronés, des rôles inappropriés, mauvais mot de passe. Pour les créations et les modifications des utilisateurs saisir des données erronées ou incomplètes dans le formulaire, pas d'accès au coffret si quelqu'un n'a pas fait la vérification des données.

Tests du Raspberry Pi avec déclenchement manuel des capteurs (gaz, mouvement) pendant un scénario de routine. Connexion SSH avec VNC Viewer, l'accès à la caméra.

7.2 Résultats obtenus

Tests de l'interface Web et de la Base de données, Cela a provoqué un rejet immédiat par le scripte JavaScript côté client. Si la vérification côté client est contournée, le script PHP prend le relais et renvoie un message d'erreur clair grâce au bloc try/catch sur la base de données.

Tests de la passerelle de paiement (Stripe). La simulation de paiements réussis et de paiements refusés (carte expirée, fonds insuffisants) via les cartes de test fournies par l'API Stripe. Le Webhook fonctionne correctement. Seuls les paiements validés déclenchent l'attribution automatique du coffret dans la base MySQL locale Always Data et l'envoi de mail de confirmation grâce à l'évènement `invoice.payment_succeeded`.

Tests de l'application d'employé. La requête SQL lié avec AND `rôle='admin'` couplée à la vérification `bcrypt` bloque systématiquement les accès non autorisés, message d'erreur quand je fais des erreurs expresses. Toutes les fonctions fonctionnent sur n'importe quel ordinateur.

Le système capte un incendie, ouvre la porte pour 10 secondes afin d'évacuer la personne qui se trouve à l'intérieur et ensuite bloque immédiatement l'ouverture des servomoteurs et insère l'alerte en temps réel dans la table des logs. Le système capte des hautes vibrations, cela ferme directement les portes. Mais pour les autres scénarios comme les vibrations, le mouvement l'alarmes s'allume qu'une seule fois sur deux. Connexion SSH fonctionnelle avec VNC Viewer, l'accès à la caméra réussit.

7.3 Bugs rencontrés

- Les autres scénarios de sécurité comme les vibrations, le mouvement l'alarmes s'allume qu'une seule fois sur deux comme cela était aléatoire.
- Lors de la création de l'exécutable final, les dépendances vers `bcrypt` et `pymysql` n'étaient pas correctement importées et installées

7.4 Corrections

Mes capteurs étaient dans une boucle `while True`, mais la boucle était si rapide qu'ils n'arrivaient pas à envoyer signal complet comme solution il fallait mettre une pause `time.sleep(s)`. Pour la création de l'exécutable je devais utiliser une commande spécifique en ligne de commande plus précise (`--hidden-import`) pour forcer l'inclusion des bibliothèques cryptographiques dans le paquetage final.

8 Conclusion

8.1 Résumé du travail réalisé

Le projet SafeBox Pro a consisté à concevoir et développer de A à Z un système de location de coffrets de haute sécurité, intégrant à la fois une dimension matérielle et logicielle. J'ai mis en place un site internet permettant aux clients de réserver et payer leur coffret via Stripe, connectée à une base de données centralisée (Always Data). Une application a été créée pour faire des vérifications nécessaires sur les clients et d'être toujours ouvert à régler leur éventuel problème. Une application de bureau en Python (Tkinter) a été développée pour permettre à l'administration de gérer le parc de coffrets et les alertes de sécurité en temps réel. Côté matériel, un Raspberry Pi gère l'ouverture physique des coffrets via RFID, keypad et autorisation du garde, le Raspberry Pi surveille l'environnement (gaz, vibrations, mouvements, température) grâce aux capteurs électroniques.

8.2 Objectifs atteints ou non

Parmi les erreurs et les difficultés durant ce projet j'ai réussi à réaliser l'essentielle, c'est-à-dire les 6 fonctionnalités du projet globale qui permettent au client d'être satisfait car j'ai garanti la fonctionnalité et la sécurité. Avec la triple authentification, je ne vois pas comment quelqu'un pourra frauder et voler quelque chose.

8.3 Compétences acquises

Grâce à ce projet j'ai appris comment fonctionne la sécurisation des mots de passe grâce à la bibliothèque Bcrypt qui peut être utilisée dans plusieurs langages de programmation et est très fiable car il est quasiment impossible de connaître le mot de passe en hash. L'utilisation de MySQL qui ressemble beaucoup à SQLite mais est dans mon cas stocker en ligne est très confortable car je pouvais vraiment lier plusieurs serveurs, périphériques ensemble. J'ai appris plus sur les tensions maximales du Raspberry Pi 4 et en vérité je ne m'attendais pas à acheter une alimentation externe, j'ai appris aussi sur les capteurs RFID, badges et le keypad Clavier

8.4 Améliorations possibles

J'aurais changé le langage de mon application, je l'aurais mis en Flask car c'est une solution plus moderne que Tkinter et je rappelle que je visais aussi dans la modernité. J'aurais plus sécurisé l'application d'employé car elle est en .exe mais il ne faut pas être super doué pour rechercher les fichiers python dans lequel se trouvent les données de ma base de données comme l'host, le nom, l'utilisateur admin et le mot de passe.

9. Bibliographie / Webographie

Langages Web (HTML, CSS, JavaScript, PHP)

PHP.net — Manuel PHP officiel, section MySQL. Disponible sur : <https://www.php.net/manual/fr/book.mysql.php> [Consulté en décembre 2025]

Kanał o wszystkim – Tutoriel complet de 26 épisodes sur le PHP Disponible sur : https://www.youtube.com/watch?v=S4WCWAOaW7c&list=PL6ackdNhY7DB04fzJz_YVyn_d6Bs9I3gvF [Consulté en décembre 2025]

Mozilla Developer Network (MDN Web Docs) — Documentation et Lesson HTML, CSS et JavaScript [en ligne]. Disponible sur : <https://developer.mozilla.org/fr/> [Consulté en décembre 2025]

PHP.net - password_hash Disponible sur <https://www.php.net/manual/en/function.password-hash.php> [Consulté en janvier 2026]

Paielement en ligne — Stripe

Shoper – Vidéo sur l'explication de Stripe disponible sur : <https://www.youtube.com/watch?v=yHSoGBgyOKA> [Consulté en 2025]

Stripe Inc. — Documentation officielle de l'API Stripe — Présentation des paiements [en ligne]. Disponible sur : <https://docs.stripe.com/payments-api/tour?locale=fr-FR> [Consulté en 2025]

Stripe Inc. — API Payment Intents — Documentation officielle [en ligne]. Disponible sur : <https://docs.stripe.com/payments/payment-intents?locale=fr-FR> [Consulté en janvier 2026]

Stripe Inc. — Introduction aux API de paiement [en ligne]. Disponible sur : <https://stripe.com/fr/resources/more/payment-apis-101> [Consulté en janvier 2026]

Hébergement Web — LWS

LWS — Site officiel LWS, hébergeur web français depuis 1999 [en ligne]. Disponible sur : <https://www.lws.fr> [Consulté en novembre 2025]

Base de données — Always Data & MySQL

Alwaysdata — Héberger son site gratuitement - Disponible sur :
<https://www.youtube.com/watch?v=SmIdDn86i5M> [Consulté en novembre 2025]

Alwaysdata — Héberger sur Always data son projet PHP - Disponible sur :

<https://www.youtube.com/watch?v=oSpbU8n7hGI> [Consulté le février 2026]

Pasja do informatyki —. 6 Cours gratuit sur la chaine Disponible sur
<https://www.youtube.com/watch?v=99JAI24Zd24&list=PLOYHgt8dIdoymv-Wzvs8M-OsKFD31VTVZ> [Consulté en Novembre 2025]

w3schools — SQL Database. Disponible sur :
https://www.w3schools.com/sql/sql_create_table.asp [Consulté en décembre 2025]

Python & Bibliothèques

Pythonguis — Build your own desktop apps with Python & Tkinter. - Disponible sur :
<https://www.pythonguis.com/tkinter-tutorial/> [Consulté en février 2026]

w3schools — Python MySQL - Disponible sur :
https://www.w3schools.com/PYTHON/python_mysql_getstarted.asp [Consulté en février 2026]

pyinstaller —. Disponible sur : <https://pyinstaller.org/en/stable/operating-mode.html> [Consulté en Mai 2026]

Raspberry Pi 4

openest.io — Intégrer un lecteur RFID à un Raspberry-Pi - Disponible sur :
<https://openest.io/prototype-raspberrypi/lecteur-rfid-raspberry-pi/> [Consulté en mai 2026]

HEO - Raspberry PI-4 consommation – Disponible sur : <https://www.hisse-et-oh.com/tavern/raspberry-pi-4-consommation> [Consulté en janvier 2026]

OpenCV — Vision par ordinateur

Wikipédia — *OpenCV* — *Bibliothèque de vision par ordinateur* [en ligne]. Disponible sur : <https://fr.wikipedia.org/wiki/OpenCV> [Consulté en avril 2026]

DataScientest — *OpenCV : tout savoir sur le principal outil de Computer Vision* [en ligne]. Disponible sur : <https://datascientest.com/opencv> [Consulté en avril 2026]

10. Annexes

Site web <https://michalowskialan.be/> (Partie intéressante)

dashboard.html

The screenshot displays the 'Mon Espace Personnel' (My Personal Space) dashboard for 'SafeBox Pro'. The page is divided into two main sections: 'Mes Informations' (My Information) and 'Nos Offres Disponibles' (Our Available Offers).

Mes Informations: This section includes a 'Modifier' (Edit) button and two calendar widgets. The first widget shows 'mai 2026' and 'Membre depuis' (Member since). The second widget shows 'P-01 Coffret' (Subscription) with an 'Annuler le coffret' (Cancel subscription) button. Below these are input fields for 'Prénom' (Jordan), 'Nom' (Micheal), 'Email' (michealbjordan23@gmail.com), and 'Téléphone' (32483037002).

Nos Offres Disponibles: This section lists three subscription options, each with a 'Choisir' (Choose) button:

- Petit Coffret:** 10x15x5 cm, 53 coffrets disponibles, 10€/mois.
- Coffret Moyen:** 20x30x15 cm, 52 coffrets disponibles, 20€/mois.
- Grand Coffret:** 40x50x30 cm, 51 coffrets disponibles, 35€/mois.

register.html

SafeBox Pro Accueil À propos Connexion

Créer un compte

Rejoignez SafeBox Pro en quelques clics

Prénom * Jean Nom * Dupont

Adresse email * michealjerdan23@gmail.com

Date de naissance * mm/dd/yyyy Numéro de GSM * 06 12 34 56 78

Adresse complète * 123 Rue de la Paix, 75001 Paris

Mot de passe * Minimum 8 caractères, 1 majuscule, 1 chiffre

Confirmer le mot de passe *

Paieiment Stripe

environnement de test Secure Box **SecureBox**

Subscribe to Coffret Moyen
€20.00 per month
Taille : 20x30x15 cm

Pay with **link** **amazon pay**

OR

Contact information
Email
email@example.com

Payment method
 Card
 Klarna

Save my information for faster checkout
Pay securely at environnement de test Secure Box and everywhere Link is accepted.

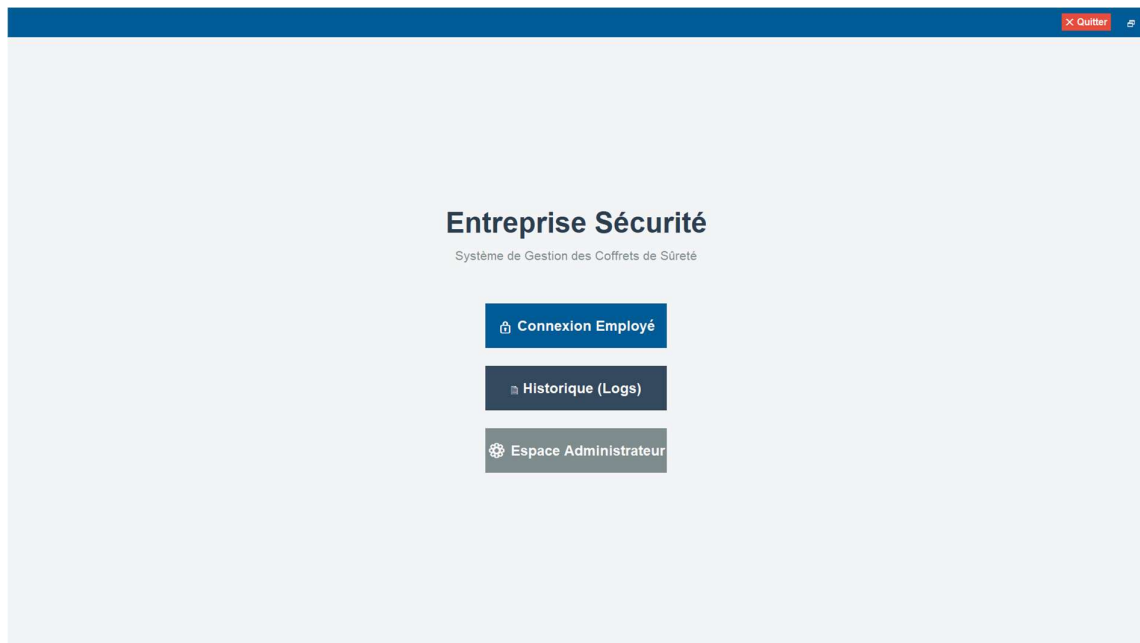
Pay and subscribe

By subscribing, you authorize environnement de test Secure Box to charge you according to the terms until you cancel.

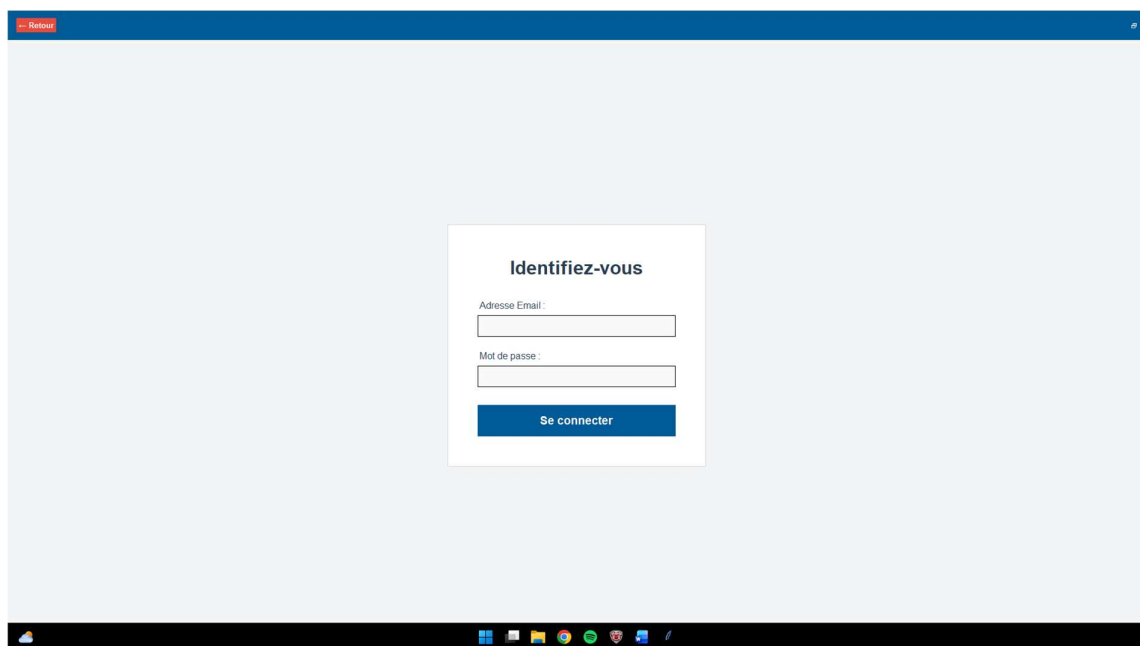
Powered by stripe | Terms | Privacy

Application d'Employé

PageAccueil.py



PageConnexion.py



PageReception.py

— Déconnexion Employé : Manon Janoz [Changer mon mot de passe](#)

■ Clients Coffrets

ID	Nom	Prénom	Email	Tel	Naissance	Adresse	Validé
1	Niemen	Czeslaw	czeslawniemen@gmail.com	+32 48545221	1965-05-04	Au. Agulter 188	<input checked="" type="checkbox"/> Oui
2	Michal	Jordan	michaljordan2@gmail.com	32483037002	1972-06-14	Rue de Neck 32, 1081, Belgique	<input checked="" type="checkbox"/> Oui
3	Messi	Lionel	lionelmessi10@gmail.com	32483037005	1989-05-08	Cuadro de sombrero 1820	<input checked="" type="checkbox"/> Oui
4	Michalowski	Alan	alanotrem@gmail.com	32483037099	2007-06-26	Rue de Neck 32, 1081, Belgique	<input checked="" type="checkbox"/> Non
5	Michalowska	Iwona	iwonamichalowska@gmail.com	32483037001	1987-07-25	Rue de Neck 32, 1081, Belgique	<input checked="" type="checkbox"/> Oui
9	Grande	Ariana	arianagrande@gmail.com	+32 482 184 14 28	1993-06-26	Au. Houzeau 87, 1180 Uccle	<input checked="" type="checkbox"/> Oui
10	Van Damme	Jean-Claude	jeanvandamme@gmail.com	+32 485 14 32 25	1960-03-25	Chau. de Saint-Job 668, 1180 Uccle	<input checked="" type="checkbox"/> Oui
11	Levandovski	Robert	robertlevandovski@gmail.com	+32 485 29 85 99	1981-01-04	Au. du Prince d'Orange 34, 1180 Uccle	<input checked="" type="checkbox"/> Oui
12	Budde	Kamil	freelucka@gmail.com	+32 483 25 93 77	1997-05-18	37, au. DeWaes Watermael, 1170 Watermael	<input checked="" type="checkbox"/> Oui
13	El Aouradi	Abdel	elabdel7152@gmail.com	+32 482 21 15 63	1965-12-08	Rue Blaes 152, 1000 Bruxelles	<input checked="" type="checkbox"/> Oui
15	De Vijf	Juliette	juliettedevijf@gmail.com	+32 483 19 41 51	2007-06-25	Rue de Saenon 72, Bruxelles, Belgique	<input checked="" type="checkbox"/> Oui
16	Decker	Elise	eliseecker@gmail.com	+32 483 95 51 18	2007-08-01	Rue Saint-Jean 47, 1000 Bruxelles	<input checked="" type="checkbox"/> Oui

Actualiser

— Déconnexion Employé : Manon Janoz [Changer mon mot de passe](#)

■ Clients Coffrets

ID	Numéro	Type	Statut	Client
1	P-01	petit	<input checked="" type="radio"/> Disponible	...
2	P-02	petit	<input checked="" type="radio"/> Loué	Ariana Grande
3	P-03	petit	<input checked="" type="radio"/> Loué	Juliette De Vijf
4	P-04	petit	<input checked="" type="radio"/> Disponible	...
5	P-05	petit	<input checked="" type="radio"/> Disponible	...
6	M-01	moyen	<input checked="" type="radio"/> Loué	Czeslaw Niemen
7	M-02	moyen	<input checked="" type="radio"/> Loué	Jean-Claude Van Damme
8	M-03	moyen	<input checked="" type="radio"/> Loué	Elise Decker
9	M-04	moyen	<input checked="" type="radio"/> Disponible	...
10	M-05	moyen	<input checked="" type="radio"/> Disponible	...
11	G-01	grand	<input checked="" type="radio"/> Loué	Lionel Messi
12	G-02	grand	<input checked="" type="radio"/> Loué	Iwona Michalowska
13	G-03	grand	<input checked="" type="radio"/> Loué	Robert Levandovski
14	G-04	grand	<input checked="" type="radio"/> Loué	Abdel El Aouradi
15	G-05	grand	<input checked="" type="radio"/> Disponible	...
16	P-01	petit	<input checked="" type="radio"/> Disponible	...
17	P-02	petit	<input checked="" type="radio"/> Disponible	...
18	P-03	petit	<input checked="" type="radio"/> Disponible	...
19	P-04	petit	<input checked="" type="radio"/> Disponible	...
20	P-05	petit	<input checked="" type="radio"/> Disponible	...
21	P-06	petit	<input checked="" type="radio"/> Disponible	...
22	P-07	petit	<input checked="" type="radio"/> Disponible	...
23	P-08	petit	<input checked="" type="radio"/> Disponible	...
24	P-09	petit	<input checked="" type="radio"/> Disponible	...
25	P-10	petit	<input checked="" type="radio"/> Disponible	...
26	P-11	petit	<input checked="" type="radio"/> Disponible	...
27	P-12	petit	<input checked="" type="radio"/> Disponible	...
28	P-13	petit	<input checked="" type="radio"/> Disponible	...
29	P-14	petit	<input checked="" type="radio"/> Disponible	...
30	P-15	petit	<input checked="" type="radio"/> Disponible	...
31	P-16	petit	<input checked="" type="radio"/> Disponible	...
32	P-17	petit	<input checked="" type="radio"/> Disponible	...
33	P-18	petit	<input checked="" type="radio"/> Disponible	...
34	P-19	petit	<input checked="" type="radio"/> Disponible	...
35	P-20	petit	<input checked="" type="radio"/> Disponible	...
36	P-21	petit	<input checked="" type="radio"/> Disponible	...
37	P-22	petit	<input checked="" type="radio"/> Disponible	...
38	P-23	petit	<input checked="" type="radio"/> Disponible	...
39	P-24	petit	<input checked="" type="radio"/> Disponible	...
40	P-25	petit	<input checked="" type="radio"/> Disponible	...
41	P-26	petit	<input checked="" type="radio"/> Disponible	...
42	P-27	petit	<input checked="" type="radio"/> Disponible	...
43	P-28	petit	<input checked="" type="radio"/> Disponible	...
44	P-29	petit	<input checked="" type="radio"/> Disponible	...

Actualiser

PageHistorique

Retour au Menu Actualiser

Journal des Actions (Logs)

ID	Date & Heure	Employé / Utilisateur	Action réalisée	Détails de l'opération
61	01/06/2026 11:16	Manon Janez	Connexion	L'employé Manon Janez s'est connecté.
60	01/06/2026 11:04	Manon Janez	Connexion	L'employé Manon Janez s'est connecté.
59	31/05/2026 12:28	Manon Janez	Connexion	L'employé Manon Janez s'est connecté.
58	30/05/2026 20:03	Manon Janez	Modification Client	Modification client ID 16
57	30/05/2026 20:03	Manon Janez	Connexion	L'employé Manon Janez s'est connecté.
56	30/05/2026 19:59	Manon Janez	Attribution Coffret	Coffre M-03 -> Client 16
55	30/05/2026 19:58	Manon Janez	Création Client	Création client : Elise Decker (Sécurisé Bcrypt)
54	30/05/2026 19:47	Manon Janez	Connexion	L'employé Manon Janez s'est connecté.
53	30/05/2026 19:45	Manon Janez	Connexion	L'employé Manon Janez s'est connecté.
52	30/05/2026 17:24	Manon Janez	Attribution Coffret	Coffre P-03 -> Client 15
51	30/05/2026 17:24	Manon Janez	Retrait Coffret	Libération coffre P-03
50	30/05/2026 17:22	Manon Janez	Attribution Coffret	Coffre P-03 -> Client 15
49	30/05/2026 17:22	Manon Janez	Création Client	Création client : Juliette De Vrij (Sécurisé Bcrypt)
48	30/05/2026 17:20	Manon Janez	Sécurité	L'utilisateur Manon Janez a modifié son MDP (Bcrypt)
47	30/05/2026 17:19	Manon Janez	Connexion	L'employé Manon Janez s'est connecté.
46	30/05/2026 17:15	Manon Janez	Connexion	L'employé Manon Janez s'est connecté.
45	30/05/2026 17:14	Manon Janez	Connexion	L'employé Manon Janez s'est connecté.
44	30/05/2026 17:12	Manon Janez	Connexion	L'employé Manon Janez s'est connecté.
43	30/05/2026 17:09	Manon Janez	Connexion	L'employé Manon Janez s'est connecté.
42	30/05/2026 17:04	Manon Janez	Connexion	L'employé Manon Janez s'est connecté.
41	30/05/2026 17:01	Manon Janez	Connexion	L'employé Manon Janez s'est connecté.
40	30/05/2026 17:00	Manon Janez	Connexion	L'employé Manon Janez s'est connecté.
39	30/05/2026 15:39	Ryan Tudor	Connexion	L'employé Ryan Tudor s'est connecté.
38	30/05/2026 15:38	Alan Michalowski	ADMIN_CREATE_USER	Création de l'employé Ryan Tudor (Compte sécurisé par hash)

PageConnexionAdmin.py

Retour

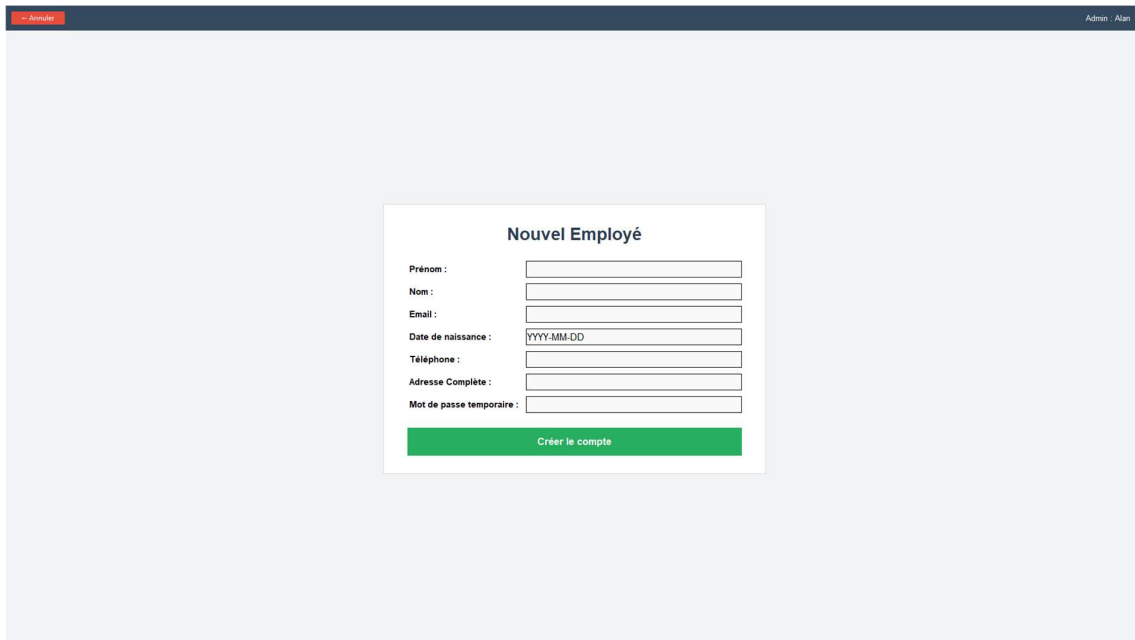
Accès Administrateur

Adresse Email :

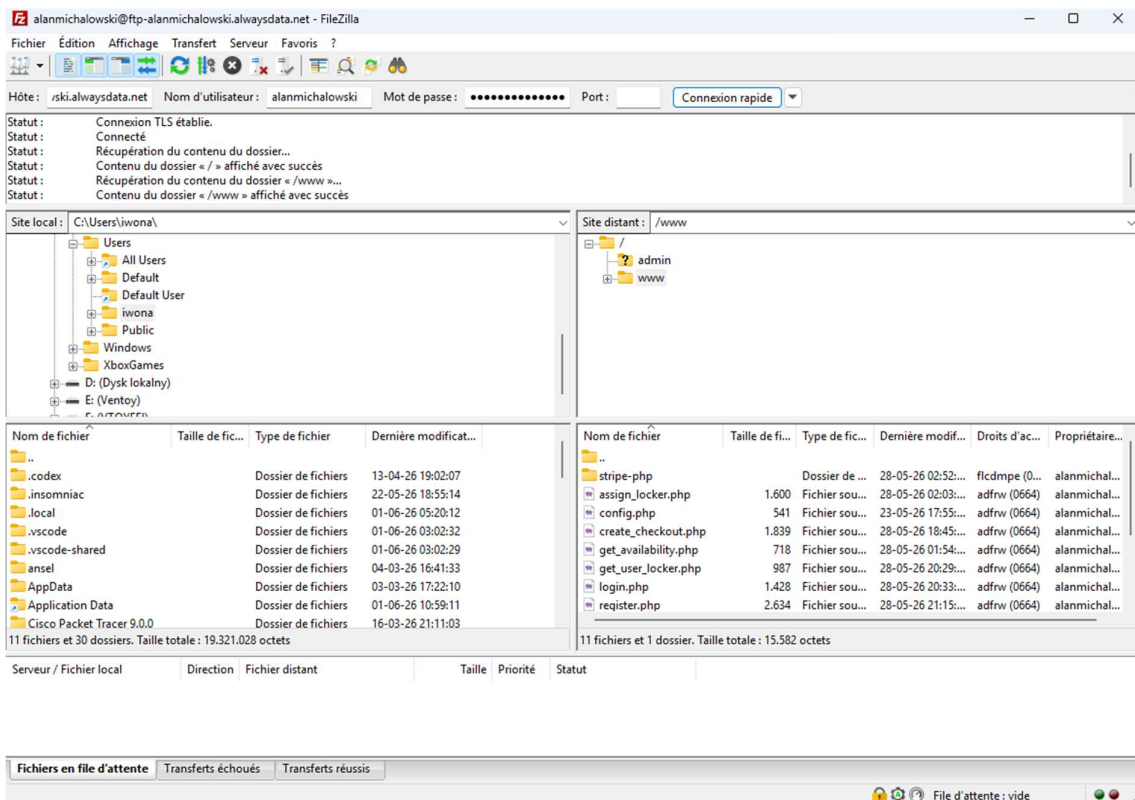
Mot de passe :

Se connecter

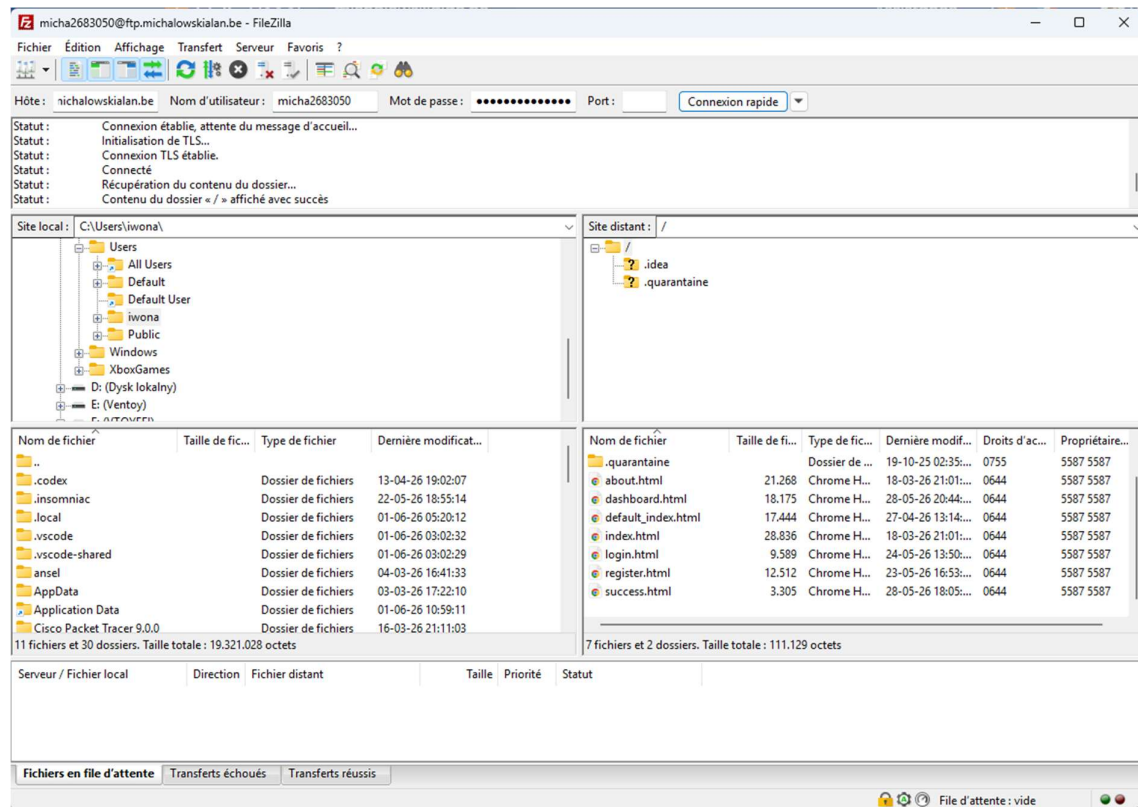
PageGestionAdmin.py



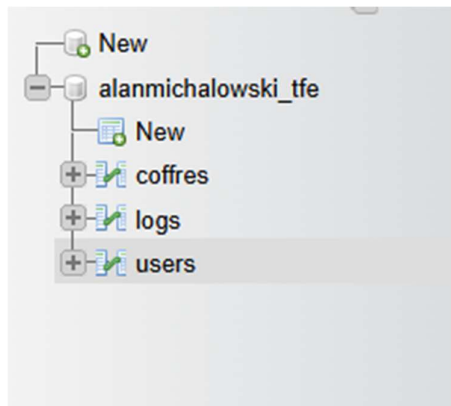
Connexion au serveur FTP Always Data où se trouve mes fichiers .PHP



Connexion au serveur FTP LWS où se trouve mes fichiers .PHP

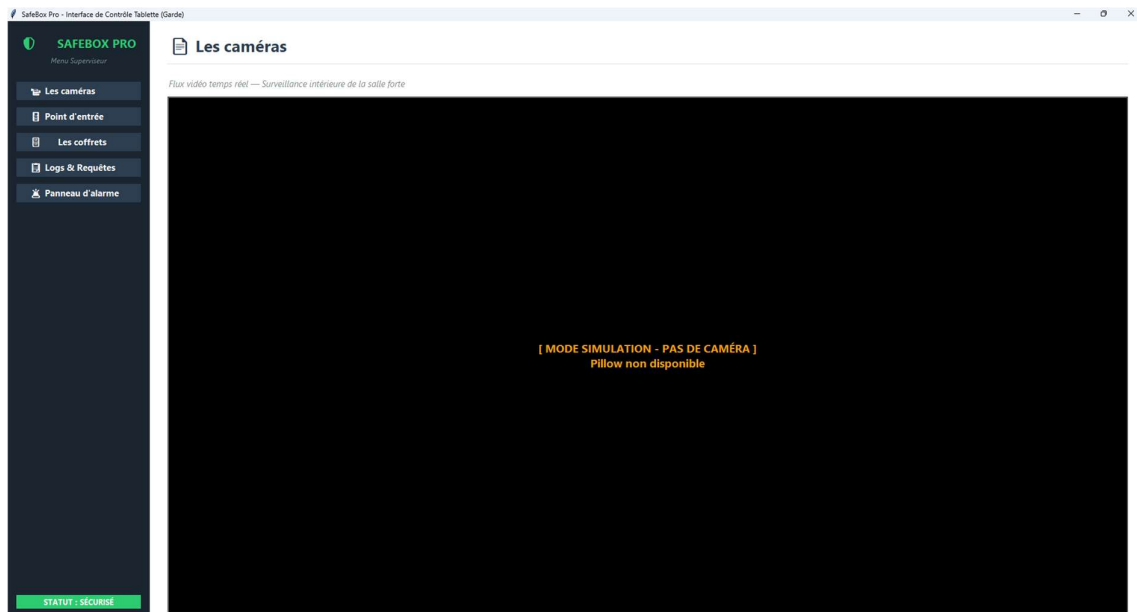


Base de données Always Data

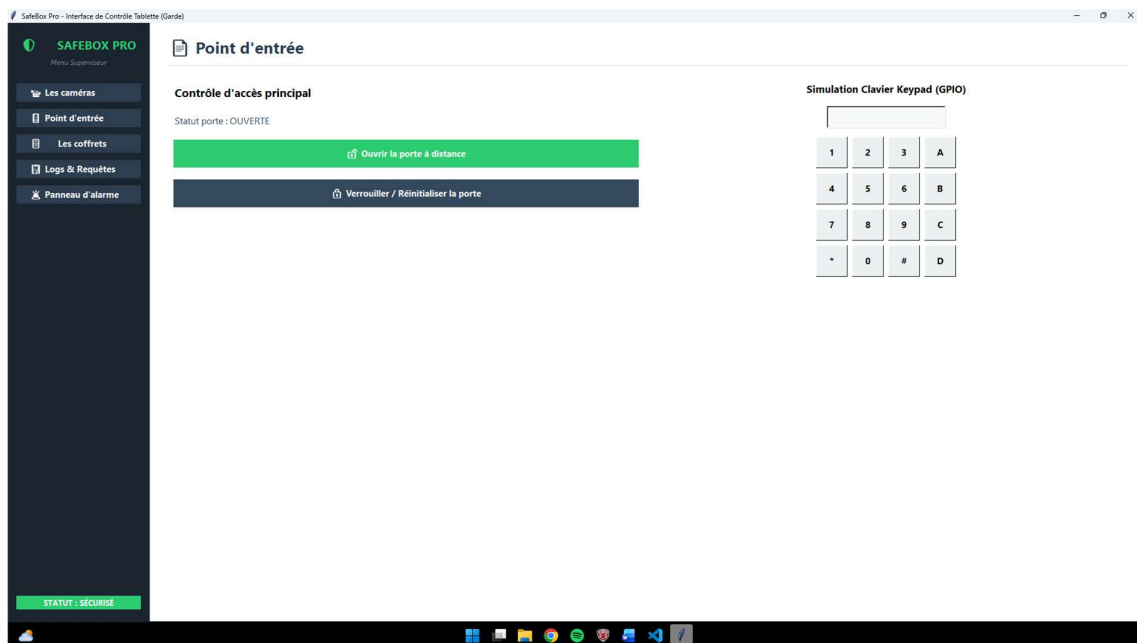


Application de sécurité

Vue sur la caméra



Ouverture de la porte



Interface d'alarmes

The screenshot displays the 'SAFEBOX PRO' alarm interface. The main title is 'Panneau d'alarme'. A green status bar at the top indicates 'AUCUNE ALARME EN COURS'. Below this, a section titled 'Métriques d'environnement en temps réel (Broches GPIO)' shows four sensor readings: Temperature (DHT22) at 21.1 °C, Humidity (DHT22) at 45.9 %, Gas Detector (MQ135) at 130 ppm, and Vibrations (GPIO 5) at Normal. A 'Panneau de simulation pour soutenance' section is also visible, containing three buttons: 'Désactiver alerte incendie artificielle', 'Désactiver capteur secousse (Séisme/Vol)', and 'Acquiescer et réinitialiser tout le système'. A sidebar on the left lists navigation options: 'Les caméras', 'Point d'entrée', 'Les coffrets', 'Logs & Requêtes', and 'Panneau d'alarme'. The bottom left corner shows 'STATUT : SÉCURISÉ'.

SAFEBOX PRO
Menu Supérieur

Les caméras
Point d'entrée
Les coffrets
Logs & Requêtes
Panneau d'alarme

STATUT : SÉCURISÉ

Panneau d'alarme

✓ AUCUNE ALARME EN COURS

Métriques d'environnement en temps réel (Broches GPIO)

Température (DHT22) 21.1 °C	Humidité (DHT22) 45.9 %	Capteur de Gaz (MQ135) 130 ppm	Vibrations (GPIO 5) Normal
--------------------------------	----------------------------	-----------------------------------	-------------------------------

Panneau de simulation pour soutenance

Provoquer une hausse anormale de gaz (MQ135)

⚠ Désactiver alerte incendie artificielle ⚡ Désactiver capteur secousse (Séisme/Vol) [E] Acquiescer et réinitialiser tout le système