

Table des matières

Contraintes du projet de PRJ1401.....	2
1. Lecteur NFC	2
2. Microcontrôleur ESP32.....	2
3. Relais.....	2
4. Loquet Magnétique	3
5. Écran LCD (avec adaptateur I2C)	3
6.Serveur	3
Tableau récapitulatif des contraintes des composants.....	4
Impact des Contraintes sur le Projet.....	4
Conclusion	4

Contraintes du projet de PRJ1401

1. Lecteur NFC

Rôle : Permet l'authentification de l'utilisateur en scannant une carte NFC.

Contraintes :

- **Alimentation :** Généralement 3,3V ou 5V, compatible avec l'ESP32.
- **Latence :** Le temps de lecture doit être rapide (<1 seconde).
- **Compatibilité :** Doit fonctionner avec les cartes NFC autorisées.
- **Sécurité :** Les données doivent être transmises de manière sécurisée pour éviter le clonage de cartes.

2. Microcontrôleur ESP32

Rôle : Cerveau du système, il reçoit les données du lecteur NFC, contrôle le loquet et affiche les informations sur l'écran.

Contraintes :

- **Alimentation :** 3,3V (nécessite un régulateur si alimenté par 5V ou plus).
- **Nombre de ports GPIO limité :** Doit gérer plusieurs périphériques sans conflit.
- **Latence :** Doit traiter les données en temps réel et envoyer rapidement les commandes.
- **Consommation électrique :** Modérée (~100-250 mA), doit être optimisée si le système est alimenté sur batterie.
- **Sécurité :** Connexions Wi-Fi/Bluetooth doivent être protégées contre les cyberattaques.

3. Relais

Rôle : Active et désactive l'alimentation du loquet magnétique en fonction des commandes de l'ESP32.

Contraintes :

- **Alimentation :** Nécessite du 5V pour fonctionner, compatible avec l'ESP32 via un transistor ou un circuit d'adaptation de tension.
- **Charge supportée :** Doit être capable de commuter la tension et l'intensité du loquet (souvent 12V ou 24V).
- **Temps de réponse :** <100 ms pour éviter tout retard dans l'ouverture.
- **Durée de vie :** Doit supporter plusieurs cycles d'activation/désactivation sans s'user rapidement.

4. Loquet Magnétique

Rôle : Verrouille et déverrouille le casier selon les instructions du relais.

Contraintes :

- **Alimentation** : Fonctionne souvent en 12V ou 24V → nécessite une alimentation adaptée.
- **Consommation électrique** : Peut consommer plusieurs ampères lorsqu'il est activé.
- **Sécurité physique** : Doit être robuste pour résister aux tentatives d'effraction.
- **Mode de verrouillage** : Certains modèles restent verrouillés même sans courant (fail-secure), d'autres se déverrouillent en cas de coupure de courant (fail-safe).
- **Temps de réaction** : Doit s'ouvrir instantanément après activation pour éviter un décalage perceptible par l'utilisateur.

5. Écran LCD (avec adaptateur I2C)

Rôle : Affiche les instructions et les statuts du casier (ex : "Casier ouvert", "Carte invalide", etc.).

Contraintes :

- **Alimentation** : Fonctionne généralement en 5V, nécessite une compatibilité avec l'ESP32 (3,3V).
- **Interface de communication** : Utilise l'I2C (moins de fils, mais nécessite des adresses spécifiques pour éviter les conflits avec d'autres périphériques).
- **Lisibilité** : Doit être visible même en conditions de faible luminosité (rétroéclairage requis).
- **Taille d'affichage** : Généralement limité à 16x2 ou 20x4 caractères → nécessite une gestion efficace des messages affichés.

6. Serveur

Rôle : L'ESP32 doit être connecté à un **serveur** contenant la **base de données** des cartes NFC autorisées. Cela ajoute des **contraintes supplémentaires** à prendre en compte :

Contraintes :

- **Connexion Wi-Fi de l'ESP32** : Permet la communication avec le serveur --> Nécessite une connexion réseau stable et sécurisée (peut être limité en cas de coupure du Wi-Fi)
- **Base de données (MySQL, Firebase, etc.)** : Stocke les identifiants NFC autorisés --> Doit être rapide pour valider les cartes NFC en temps réel (latence minimale)
- **Sécurité des échanges** : Protège contre les accès non autorisés --> Doit utiliser le chiffrement des données (HTTPS, authentification sécurisée)
- **Synchronisation** : Mise à jour en temps réel des accès --> Assurer que la base de données reste à jour sans latence excessive
- **Serveur hébergé (local ou cloud)** : Gère les requêtes et les accès aux casiers --> Si en cloud, dépend d'Internet ; si local, doit être robuste et accessible en interne

Tableau récapitulatif des contraintes des composants

Composant	Rôle	Contraintes Principales
Lecteur NFC	Lecture des cartes NFC/RFID pour identifier l'utilisateur	⚡ Alimentation : en 3.3V ou 5V (doit être compatible avec l'ESP32) ⚡ Temps de réponse : rapide (<1 seconde) pour éviter toute latence ⚡ Sécurité : Transmission sécurisée des données pour éviter le donage de cartes
ESP32 Devkit V4	Microcontrôleur principal qui gère les entrées/sorties et la communication avec le serveur	⚡ Alimentation 3.3V : Nécessite une conversion si utilisé avec des composants 5V ou 12V ⚡ Nombre limité de GPIO : Doit gérer plusieurs périphériques sans conflits ⚡ Connexion Wi-Fi : Essentielle pour accéder à la base de données distante, mais vulnérable aux interruptions réseau ⚡ Sécurité : Doit chiffrer les échanges avec le serveur pour éviter toute interception (HTTPS, MQTT avec chiffrement) ⚡ Latence : Doit traiter les données en temps réel pour une ouverture instantanée des casiers ⚡ Consommation d'énergie : Peut être un facteur important si alimenté par batterie
Relais 5V	Active et désactive l'alimentation du loquet magnétique	⚡ Incompatibilité de tension avec l'ESP32 (3.3V) → nécessite un circuit d'adaptation (ex : transistor) ⚡ Durée de vie : limitée à un certain nombre de cycles (usure mécanique à long terme) ⚡ Temps de commutation : Doit être <100 ms pour éviter tout retard dans l'ouverture du casier
Loquet Magnétique 12V/24V	Verroille et déverrouille le casier en fonction des commandes du relais	⚡ Alimentation : Fonctionne en 12V ou 24V → nécessite une alimentation séparée ⚡ Consommation électrique élevée : Peut atteindre plusieurs ampères en activation ⚡ Mode de verrouillage : Certains loquets restent verrouillés sans courant (fail-secure), d'autres se déverrouillent en cas de coupure de courant (fail-safe) ⚡ Sécurité physique : Doit être robuste contre les tentatives d'effraction ⚡ Temps de réaction : Doit s'ouvrir instantanément après activation pour éviter une latence perceptible
Écran LCD I2C	Affichage des instructions et statuts du casier (ex: "Carte valide", "Accès refusé")	⚡ Alimentation 5V : Nécessite un convertisseur de niveau logique si l'ESP32 fonctionne en 3.3V ⚡ Utilisation de l'I2C : Adresse spécifique pour éviter les conflits avec d'autres périphériques ⚡ Lisibilité : Doit être lisible dans toutes les conditions de lumière (rétroéclairage nécessaire) ⚡ Taille limitée : Généralement 16x2 ou 20x4 caractères, il faut optimiser les messages affichés
Connexion Wi-Fi	Permet à l'ESP32 de communiquer avec le serveur qui gère les autorisations NFC	⚡ Fiabilité du réseau : Peut être affectée par la couverture Wi-Fi et les interférences ⚡ Sécurité : Doit être chiffrée (WPA2 au minimum) pour éviter toute attaque man-in-the-middle ⚡ Gestion des pannes réseau : Le système doit prévoir un mode dégradé si la connexion est perdue (ex: cache local des cartes NFC autorisées)
Base de données (MySQL, Firebase, etc.)	Stocke les cartes NFC autorisées et gère l'accès aux casiers	⚡ Latence minimale : L'accès à la base doit être rapide pour éviter un retard dans l'ouverture des casiers ⚡ Mise à jour en temps réel : Assurer que les nouveaux utilisateurs ou modifications des accès sont immédiatement pris en compte ⚡ Sécurisation des données : Les échanges entre l'ESP32 et la base doivent être chiffrés (HTTPS, SSL/TLS) pour éviter toute fuite d'informations ⚡ Mode hors-ligne : Une solution doit être prévue pour un fonctionnement temporaire sans connexion au serveur
Serveur hébergeant la base de données	Reçoit les requêtes de l'ESP32 et vérifie si la carte NFC est autorisée	⚡ Disponibilité : Doit être en ligne en permanence pour éviter les interruptions ⚡ Capacité de gestion simultanée : Doit pouvoir traiter plusieurs requêtes en même temps sans ralentissement ⚡ Type d'hébergement : Serveur local (nécessite maintenance) ou cloud (dépendant d'Internet) ⚡ Temps de réponse : Doit répondre en quelques millisecondes pour un fonctionnement fluide

Impact des Contraintes sur le Projet

- **Optimisation des échanges serveur-ESP32** pour limiter la latence (utilisation d'un protocole léger comme MQTT).
- **Sécurisation des communications** pour éviter les failles de sécurité.
- **Prévoir un mode de secours** en cas de panne réseau (ex: cache local des cartes NFC).
- **Assurer une alimentation stable** pour les composants énergivores comme le loquet magnétique.

Conclusion

Chaque composant du système a ses propres **contraintes techniques, énergétiques et sécuritaires**. Il est essentiel de bien les gérer pour assurer la **fiabilité et la fluidité du système**.