Dossier Technique du projet « Accès campus » - Partie gestion des données

Table des matières

[I – Présentation du projet 1](#_Toc194572553)

[1.1 - Contexte et objectifs 1](#_Toc194572554)

[1.2 - Description du système 2](#_Toc194572555)

[1.3 – Fonctionnement 2](#_Toc194572556)

[1.4 - Architecture du système 3](#_Toc194572557)

[II – Spécification techniques 4](#_Toc194572558)

[2.1 – Base de données 4](#_Toc194572559)

[2.3 Technologies utilisées 7](#_Toc194572560)

[III- Production 7](#_Toc194572561)

[1.1 Premier incrément 7](#_Toc194572562)

# 

# I – Présentation du projet

## 1.1 - Contexte et objectifs

Le projet Campus Accès vise à améliorer la gestion et la sécurité des accès aux salles du Campus Saint Aubin La Salle. Grâce à un système automatisé basé sur des badges RFID, les étudiants et le personnel peuvent accéder aux salles de manière sécurisée tout en enregistrant leurs entrées et sorties. L’objectif est d’assurer un contrôle centralisé des accès, optimisé grâce au Poste Serveur Web (PSW), qui héberge la base de données PostgreSQL et l’API FastAPI pour traiter les demandes en temps réel.



## 1.2 - Description du système

Le Poste Serveur Web (PSW) est au cœur du système, assurant la gestion centralisée des accès. Il héberge la base de données PostgreSQL, qui stocke les informations des utilisateurs, des badges et des historiques d’accès. L’API FastAPI, développée en Python, permet d’interagir avec la base et de fournir une interface accessible aux autres composants du projet. Les Poignées Électroniques Autonomes (PEA) communiquent avec le PSW pour vérifier les accès, les Bornes d’Appel Etudiant automatise l’appel dans un cours, tandis que le Poste de Gestion des Salles (PGS) permet d’administrer les réservations et les droits d’accès.

Une image contenant texte, capture d’écran, diagramme, Police

Le contenu généré par l’IA peut être incorrect.Une image contenant texte, capture d’écran, diagramme, ligne

Le contenu généré par l’IA peut être incorrect.

## 1.3 – Fonctionnement

Lorsqu’un étudiant scanne son badge sur une PEA, l’API FastAPI envoie une requête à la base de données PostgreSQL, qui vérifie si l’utilisateur est autorisé. Si l’accès est validé, la porte s’ouvre et l’entrée est enregistrée. En cas de refus, une alerte est envoyée à l’administrateur. Ce dernier utilise le PGS pour modifier les permissions et gérer les réservations des salles. Toutes les interactions sont centralisées sur le PSW, qui stocke et analyse les données d’accès en temps réel.Une image contenant texte, diagramme, ligne, capture d’écran

Le contenu généré par l’IA peut être incorrect.

## 1.4 - Architecture du système

Le PSW repose sur une architecture en plusieurs couches, garantissant un traitement efficace des requêtes. L’interface utilisateur, accessible depuis un navigateur web, interagit avec l’API FastAPI, qui exécute les requêtes et communique avec la base de données PostgreSQL. Le serveur Debian 11, hébergeant l’ensemble, assure la disponibilité du service et la gestion des accès en continu. Ce modèle permet une réponse rapide et sécurisée aux demandes des PEA et du PGS.

Une image contenant texte, capture d’écran, diagramme, conception

Le contenu généré par l’IA peut être incorrect.

# II – Spécification techniques

## 2.1 – Base de données

Les données sont organisées de la sorte :

Une image contenant diagramme, ligne, texte, Plan

Le contenu généré par l’IA peut être incorrect.

Il m’a fallut plusieurs essaie pour atteindre ce modèle. Les autres versions ainsi que le script SQL sont disponibles sur notre github.