Dossier Technique du projet « Accès campus » - Partie PSW

Table des matières

[I – Présentation du projet 1](#_Toc193663696)

[1.1 - Contexte et objectifs 1](#_Toc193663697)

[1.2 - Description du système 2](#_Toc193663698)

[1.3 – Fonctionnement 2](#_Toc193663699)

[1.4 - Architecture du système 3](#_Toc193663700)

[II – Spécification techniques 4](#_Toc193663701)

[2.1 – Base de données 4](#_Toc193663702)

[2.2 Contraintes et exigences 6](#_Toc193663703)

[2.3 Technologies utilisées 6](#_Toc193663704)

[III- Production 7](#_Toc193663705)

[1.1 Premier incrément 7](#_Toc193663706)

# 

# 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, tandis que le Poste de Gestion des Salles (PGS) permet d’administrer les réservations et les droits d’accès. L’ensemble fonctionne sur un serveur Debian 11, garantissant une exécution stable et sécurisée.

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

Le MCD définit les différentes entités et relations qui composent la base de données du PSW. Chaque utilisateur est relié au maximum à un badge, il peut sois être un élève, un enseignant, un personnel, un invité, ou alors un administrateur. L’accès aux salles est tracé grâce à la table Log qui enregistre les passages. Cette table sert aussi à notifier les absences et les retards en comparant le lieu et l’horaire avec la table Creneau\_EDT\_Elève. Ils sont ensuite enregistrés dans des tables à leur nom. Les différentes tables « Creneau\_EDT » servent à l’affichage des emplois du temps sur le PSW. La base de données est donc relationnelle donc il est tout logique de partir sur du langage SQL.

Voici une évolution de la représentation du MCD de la base de données :

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

Le contenu généré par l’IA peut être incorrect.

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

Le contenu généré par l’IA peut être incorrect.

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

Le contenu généré par l’IA peut être incorrect.

Une image contenant diagramme, ligne, texte, Plan

Le contenu généré par l’IA peut être incorrect.

## 2.2 Contraintes et exigences

La base de données doit être capable de traiter plus de 2000 badges et 300 salles en garantissant une disponibilité 24/7. La base de données PostgreSQL doit être optimisée pour supporter un grand nombre de requêtes simultanées, avec un temps de réponse inférieur à 500 ms. La sécurité des données est une priorité, avec une gestion rigoureuse des permissions et un chiffrement des accès. L’ensemble du système doit fonctionner de manière fiable, sans interruption, pour assurer une gestion fluide et sécurisée des accès au campus.

## 2.3 Technologies utilisées

Le PSW est basé sur des technologies robustes et performantes. L’API backend est développée en Python avec FastAPI, garantissant une gestion rapide et asynchrone des requêtes. La base de données PostgreSQL, optimisée pour gérer un grand nombre d’entrées, assure un stockage sécurisé des informations d’accès. L’ensemble fonctionne sur Debian 11, un système d’exploitation reconnu pour sa stabilité. Grâce à l’intégration des badges RFID, la gestion des accès est fluide et automatisée, réduisant ainsi la charge administrative.

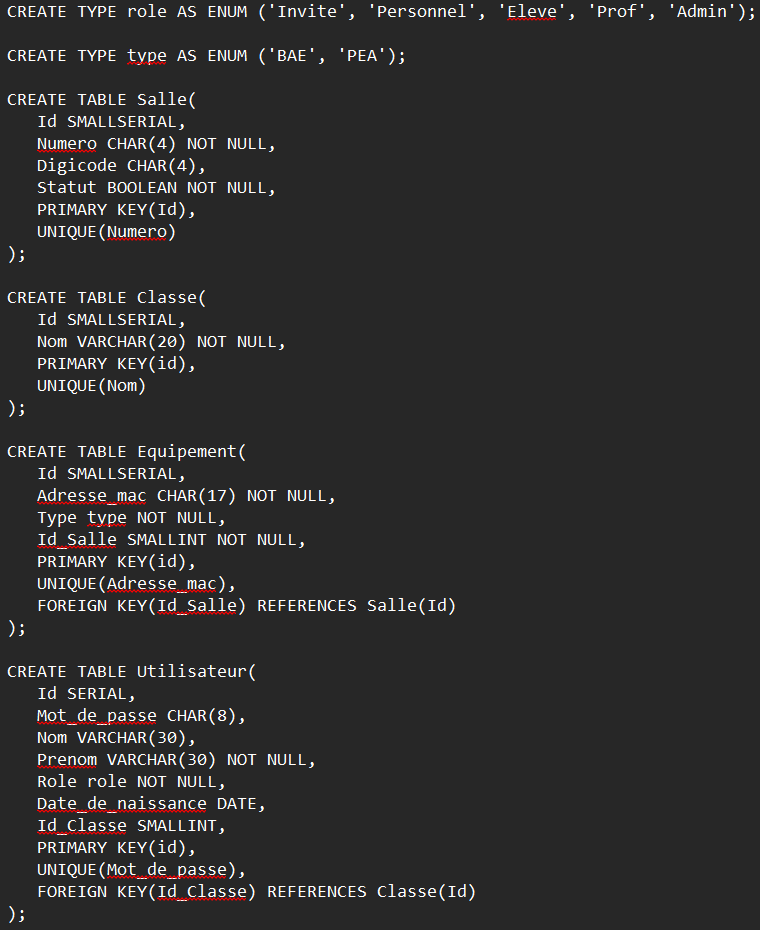
# III- Production

## Premier incrément

Pour mon premier incrément je mettre en place ma base de données et permettre l’accès aux données pour les premiers incréments de mes collègues.

Je dois donc ajouter les tables EDTUtilisateur, Equipement, Utilisateur, Badge, Salle, Classe, Absence, Retard et Autorisation.

Voici donc le script SQL permettant l’ajout de ces tables dans PostgreSQL :



Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Ensuite je code l’API avec FastAPI avec l’organisation suivante :

/API

- main.py

- schemas.py

- models.py

- database.py

/routes

- salle.py

- classe.py

- etc…