Annexes du projet B-Formation

Toute la documentation du projet est aussi disponible avec les fichiers originels sur le repo github correspondant au projet : [Cliquez ici](https://github.com/NathanChevrollier/B-formation/tree/main/Documentation%20B-form)

(https://github.com/NathanChevrollier/B-formation)

# Sommaire

Annexe 0 – Cahier des charges

Annexe 1 – Dictionnaire de données

Annexe 2 – MCD (Modèle Conceptuel de Données)

Annexe 3 – MLD (Modèle Logique de Données)

Annexe 4 – MPD (Modèle Physique de Données)

Annexe 5 – Diagramme de Gantt

Annexe 6 – Diagramme de contexte

Annexe 7 – Diagramme de package

Annexe 8 – Diagramme de classes

Annexe 9 – Cartographie technique

# Annexe 0 – Cahier des charges

B-Formation Cahier des charges

**Contexte**

Le client nous a mandaté pour développer une application web de gestion des classes et des présences intégrant un système de validation des présences via un bouton de signature. L’application doit permettre de centraliser la gestion des utilisateurs (enseignants et élèves), des matières, des classes et des emplois du temps, ainsi que la validation des présences, avec une traçabilité précise grâce à l’enregistrement de l’horodatage.

**Objectif**

Fournir une solution complète qui :

Gère efficacement les utilisateurs (administrateurs, enseignants, élèves) et leurs droits d’accès.

Organise les classes, matières et emplois du temps de manière intuitive.

Permet la validation des présences par un bouton de signature visible uniquement lorsque l’enseignanta confirmé la présence de l’élève, et enregistre un horodatage dans la base de données.

**Public cible**

Établissements d’enseignement et responsables pédagogiques souhaitant moderniser et sécuriser la gestion des classes et des présences.

**Expression des besoins**

Le client souhaite disposer d’une application permettant de :

Assurer une gestion centralisée et optimisée des utilisateurs et des classes.

Offrir une interface conviviale et sécurisée pour la gestion des emplois du temps et des affectations.

Intégrer un mécanisme de validation des présences basé sur un bouton de signature (affichéuniquement après confirmation de l’enseignant) qui, lorsqu’il est cliqué, enregistre automatiquementl’horodatage dans la base de données.

**Fonctionnalités MVP (Minimum Viable Product)**

* + - 1. **Gestion des utilisateurs :**Création, modification et suppression des comptes pour administrateurs, enseignants etélèves.
    1. Authentification sécurisée (email et mot de passe chiffré) avec gestion des rôles.
    2. **Gestion des classes :**Création, modification et suppression des classes.

Affectation des enseignants et des élèves aux classes.

* + 1. **Gestion des matières et emplois du temps :**Configuration des matières et définition des créneaux horaires par classe.

Consultation de l’emploi du temps via un espace personnalisé.

* 1. **Validation des présences :**Mise en place d’un bouton de signature, visible uniquement lorsque l’enseignant a confirmé laprésence.
  2. Enregistrement de l’horodatage du clic dans la base de données pour assurer la traçabilité.

**Fonctionnalités MLP (Minimum Lovable Product)**

Visualisation en temps réel des présences et des signatures.

Interface responsive et design moderne grâce à Bootstrap.

Historique détaillé des signatures et possibilité de filtrer les données par classe ou par période.

**Contraintes techniques**

* + 1. Développement : PHP orienté objet (POO) afin d’assurer une structure modulaire et évolutive.

Base de données : MySQL avec une modélisation relationnelle robuste (MCD, MLD, MPD).

Environnements : **Local :** Développement sous XAMPP (Apache, PHP, MySQL, PhpMyAdmin).

**Production :** Déploiement sur un VPS OVH avec nginx et gestion via PhpMyAdmin.

Sécurité : Gestion sécurisée des sessions et des authentifications (cryptage avec password\_hash()).

Utilisation de HTTPS et mise en place d’un certificat SSL pour le déploiement.

Déploiement : Déploiement sur un nom de domaine avec intégration continue des mises à jour.

**Template et Essais**

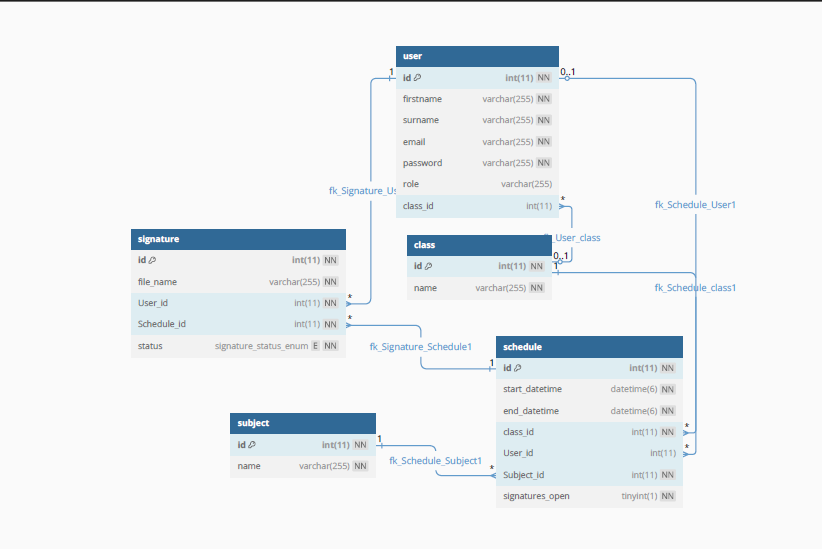
Interface utilisateur : Utilisation de Bootstrap pour garantir une interface responsive et ergonomique.

Validation des fonctionnalités : Réalisation d’essais manuels pour confirmer le bon fonctionnement de l’application (authentification, gestion des classes, validation des présences, etc.).

# Annexe 1 – Dictionnaire de données

# 

# Annexe 2 – MCD (Modèle Conceptuel de Données)



# Annexe 3 – MLD (Modèle Logique de Données)

CREATE TABLE CLASS (id INT NOT NULL,name VARCHAR(255) NOT NULL,PRIMARY KEY (id));

CREATE TABLE SUBJECT (id INT NOT NULL,name VARCHAR(255) NOT NULL,PRIMARY KEY (id));

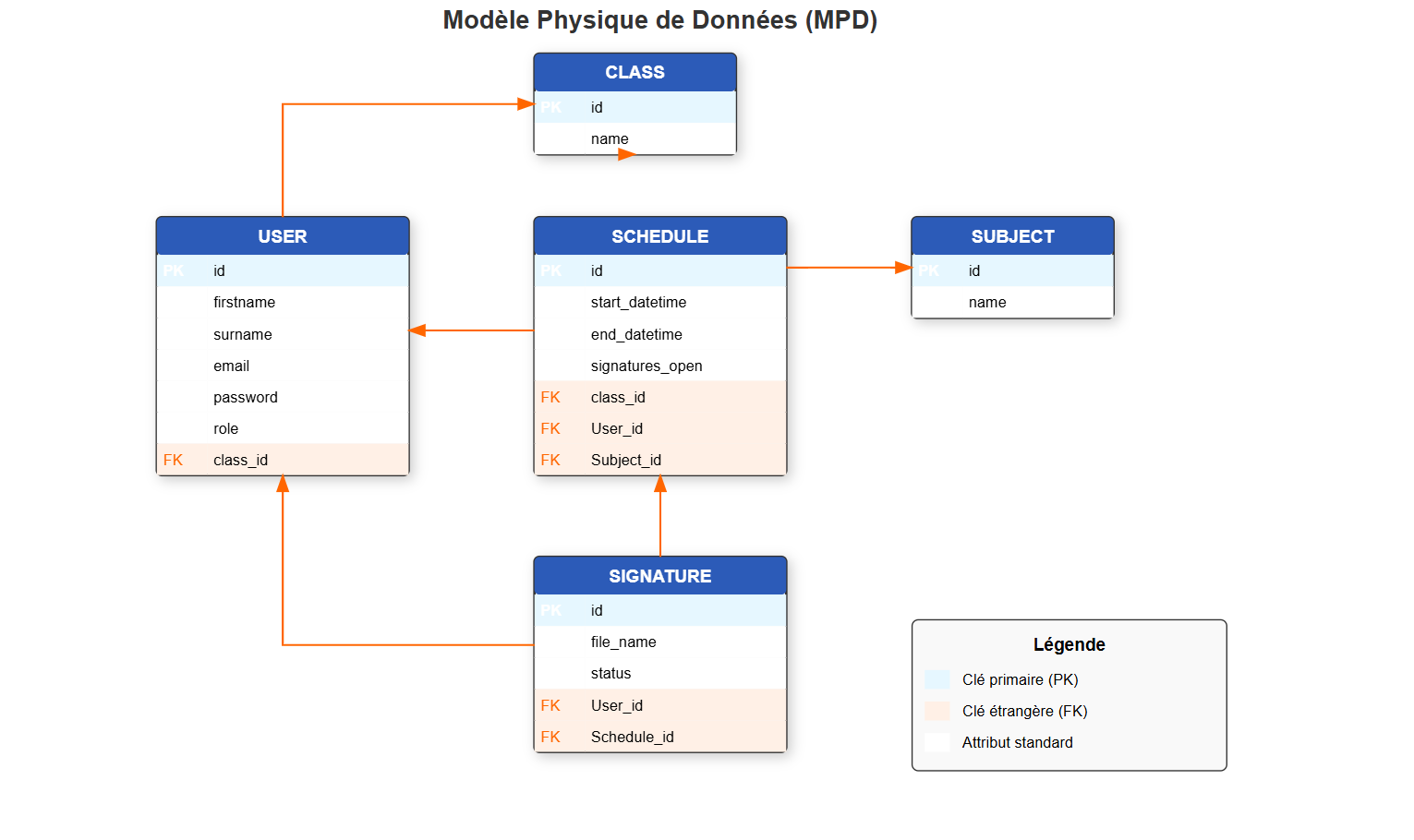
CREATE TABLE USER (id INT NOT NULL,firstname VARCHAR(255) NOT NULL,surname VARCHAR(255) NOT NULL,email VARCHAR(255) NOT NULL,password VARCHAR(255) NOT NULL,role VARCHAR(255),class\_id INT,FOREIGN KEY (class\_id) REFERENCES CLASS(id),PRIMARY KEY (id));

CREATE TABLE SCHEDULE (id INT NOT NULL,start\_datetime DATETIME(6) NOT NULL,end\_datetime DATETIME(6) NOT NULL,signatures\_open TINYINT(1) NOT NULL DEFAULT 0,class\_id INT NOT NULL,User\_id INT,Subject\_id INT NOT NULL,FOREIGN KEY (class\_id) REFERENCES CLASS(id),FOREIGN KEY (User\_id) REFERENCES USER(id),FOREIGN KEY (Subject\_id) REFERENCES SUBJECT(id),PRIMARY KEY (id));

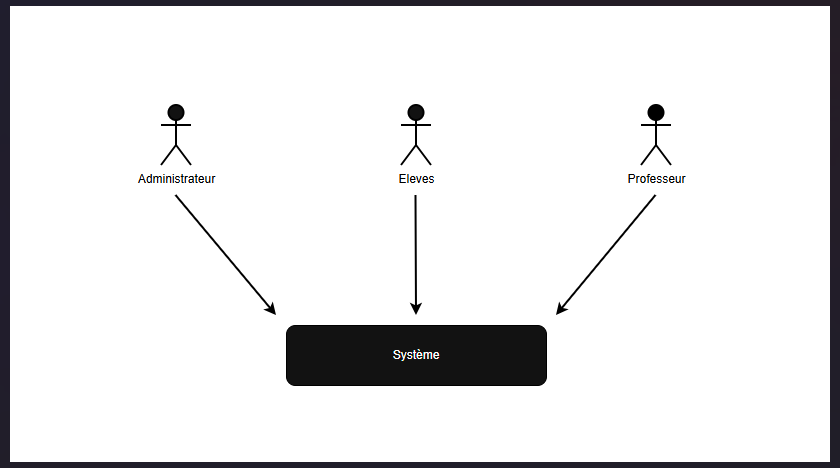
CREATE TABLE SIGNATURE (id INT NOT NULL,file\_name VARCHAR(255) NOT NULL,

status ENUM('pending', 'validated', 'absent') NOT NULL DEFAULT 'pending',User\_id INT NOT NULL,Schedule\_id INT NOT NULL,FOREIGN KEY (User\_id) REFERENCES USER(id),FOREIGN KEY (Schedule\_id) REFERENCES SCHEDULE(id),PRIMARY KEY (id));

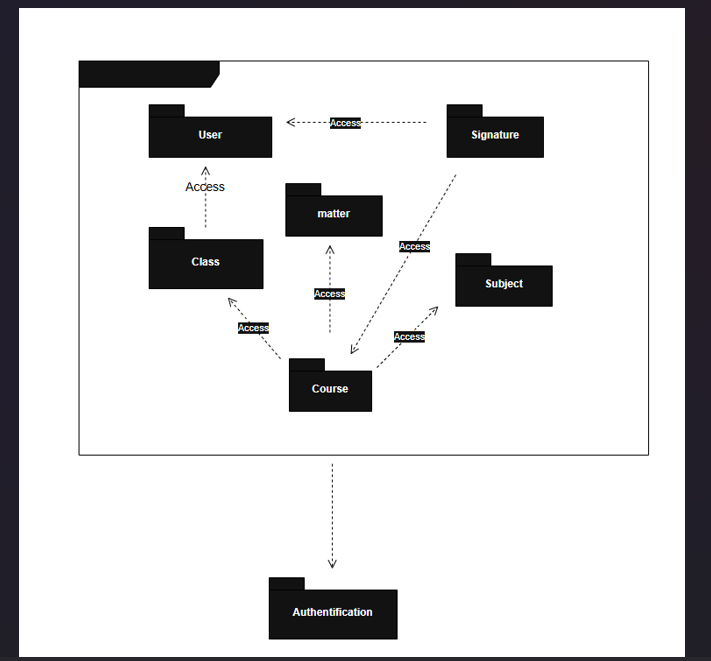
# Annexe 4 – MPD



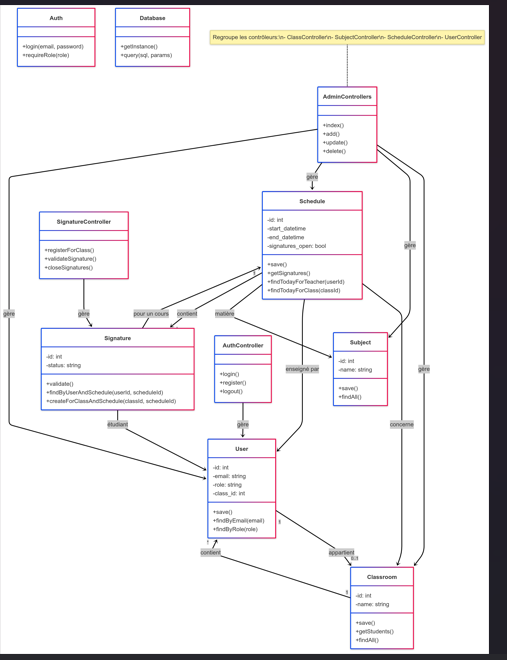
# Annexe 5 - Diagramme de context



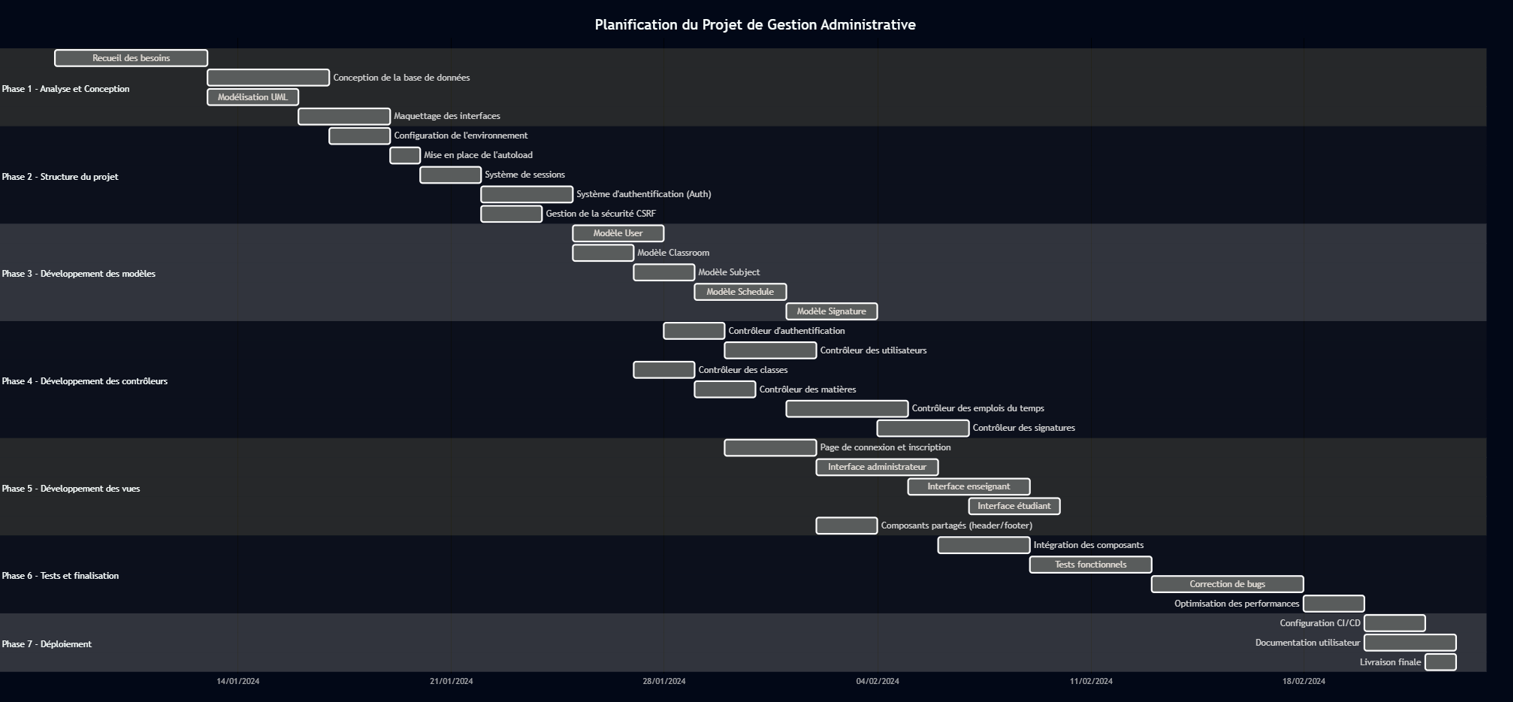
# Annexe 6 - Diagramme de package



# Annexe 7 - Diagramme de classes



# Annexe 8 - Diagramme de Gantt



# Annexe 9 - Cartographie technique

