2SIO

Wissem Laffi

Yanis Bouquet

Alexis Beriot

Saadia Zakariyya

***Spécifications Technico***

***Fonctionnelles***

***Détaillées***

Sommaire

[I. Description du projet 2](#_Toc182491137)

[II. Modélisation UML 4](#_Toc182491138)

[III. Ecran 5](#_Toc182491139)

[IV. Base de données 15](#_Toc182491140)

[1- Dictionnaire de données 15](#_Toc182491141)

[2- Dépendances fonctionnelles 15](#_Toc182491142)

[3- Modèle Conceptuel de Données (MCD) 16](#_Toc182491143)

[4- Structure des tables 18](#_Toc182491144)

[V. Technique 19](#_Toc182491145)

[VI. Gestion projet 20](#_Toc182491146)

[VI.I Trello 25](#_Toc182491147)

[VI.II Planning 26](#_Toc182491148)

[VI. Annexes 28](#_Toc182491149)

# I. Description du projet

L'école FSI à Lyon propose des formations en alternance jusqu'au BAC+5, notamment une filière numérique. Le suivi des étudiants en alternance est actuellement géré par un fichier Excel, ce qui pose des problèmes de performance et de sécurité. La coordinatrice du tutorat souhaite donc développer une application web pour faciliter le suivi des étudiants et des tuteurs.

La gestion du tutorat sera centralisée dans une Base de Données (BDD) dédiée. Cette BDD stockera l'ensemble des informations relatives aux étudiants, aux tuteurs et au suivi des sessions de tutorat.

Le projet vise à développer une application web pouvant gérer le suivi des élèves ainsi que des tuteurs au sein d’un établissement scolaire. Cette application web sera divisée en plusieurs parties :

* Une page inscription / connexion pour les élèves et les tuteurs. Pour les tuteurs, permettre l’ajout de notes, de suivi des élèves…Pour les élèves, consultation de leurs notes
* Une partie connexion pour la coordinatrice des tuteurs. Ajout de nouveaux élèves, de tuteurs, affecter les tuteurs aux élèves.
* Une page alerte, permettant de savoir les retards, absences des élèves.
* Une partie paramètre, permettant de définir des informations telles que la spécialité, la classe et le nom et prénom des tuteurs.

L’application web aura comme nom : eFsi

Le projet sera développé par 3 développeurs. Un développeur s’occupera du design du site web et l’autre s’occupera de son bon fonctionnement. Le 3e développeur s’occupera de la base de données, de la connexion pour les tuteurs et étudiants. Il y aura également le chef de projet qui s’occupera des documents professionnels, de certaines pages et de l’avancement du projet.

Le site internet doit respecter les conditions du client qui sont :

* Un site sobre
* Respect du MCD
* Le base de données doit être fait en SQL sous MySQL et l’application web en PHP
* Chaque front doit être cohérent avec les autres.
* Le logo doit apparaître à chaque page
* L’application doit être responsive, c’est-à-dire qu’elle doit fonctionner sur plusieurs types d’appareils (téléphone, ordinateur, tablette, …).
* L’application web doit être sécurisé
* Le MVC doit être suivi

# Modélisation UML

Pour ce projet, une modélisation préalable est nécessaire afin de définir les liens entre les différentes classes du système. Nous utiliserons un diagramme UML pour représenter cette structure.

Pour la réalisation de ce diagramme, nous avons choisi WhiteStarUML, un logiciel familier pour la création de diagrammes de classe.

**Structure du diagramme :**

Le diagramme de classe présente une hiérarchie claire entre les entités du système.

* **Héritage :**
  + Les classes **Bilan1** et **Bilan2** héritent de la classe **Bilan**. Cette relation est justifiée par la présence d'attributs communs aux deux classes, à l'exception du sujet de mémoire.
  + Les classes **Administrateur**, **Tuteur** et **Etudiant** héritent de la classe **Utilisateur**. Cette structure logique permet de regrouper les attributs et méthodes communs à tous les types d'utilisateurs.

Cette approche permet de garantir une organisation claire et cohérente du système, facilitant ainsi le développement et la maintenance du site web.

A screenshot of a computer screen

Description automatically generated

# Ecran

**Palette des Couleurs**

|  |  |  |
| --- | --- | --- |
| Couleur | Code Hexadécimal | Utilisation |
|  | FFFDFD | Fond |
|  | 9ED6A7 | Accentuation 1 |
|  | D9D9D9 | Accentuation 2 |
|  | 000000 | Texte |
|  | 5031FF | Hyperlien |

Pour ce site web, nous utiliserons un fond blanc et vert, avec les textes en noir et le logo du site sur la gauche. Voici notre maquette :

**Page de Connexion**

A screenshot of a login form

Description automatically generated

**Page d’Accueil (Etudiant)**

A screenshot of a computer

Description automatically generated

**Page Mes Infos (Etudiant)**

A screenshot of a phone

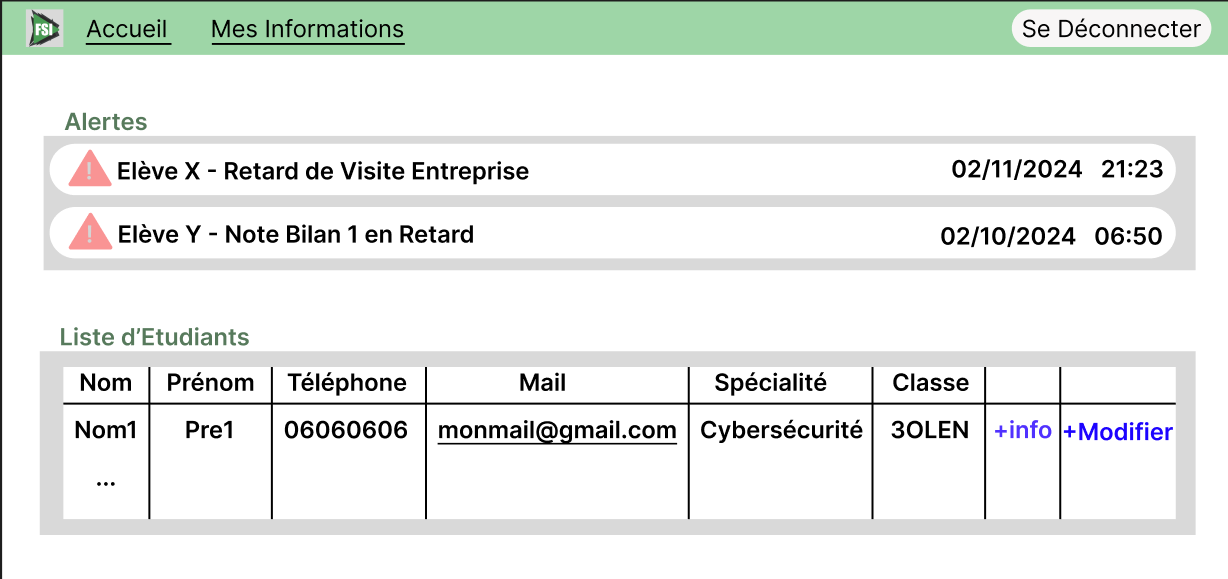
Description automatically generated

**Page Bilan**

A screenshot of a computer

Description automatically generated

**Page d’Accueil (Tuteur)**



**Page Mes Infos (Tuteurs)**

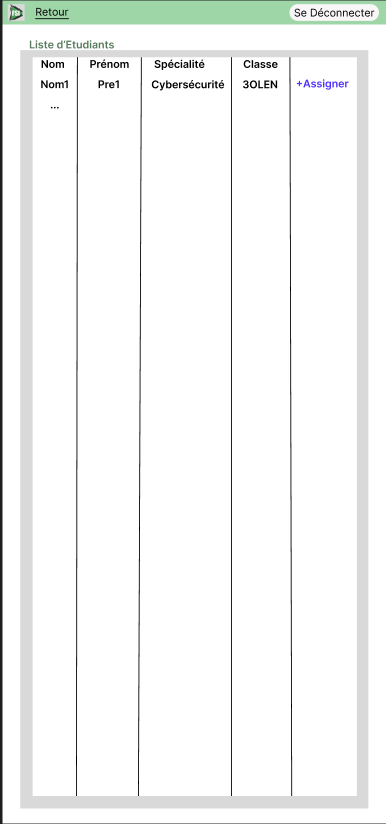
A screenshot of a computer

Description automatically generated

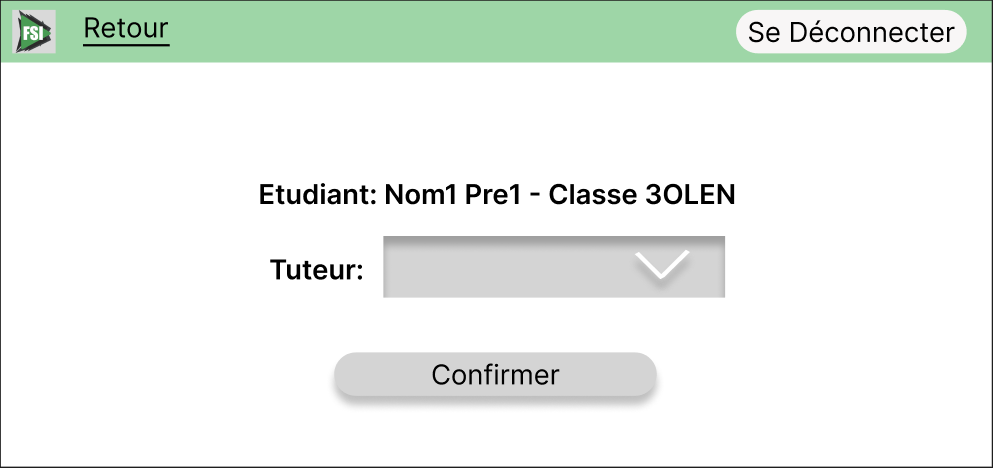
**Page d’accueil administrateur**



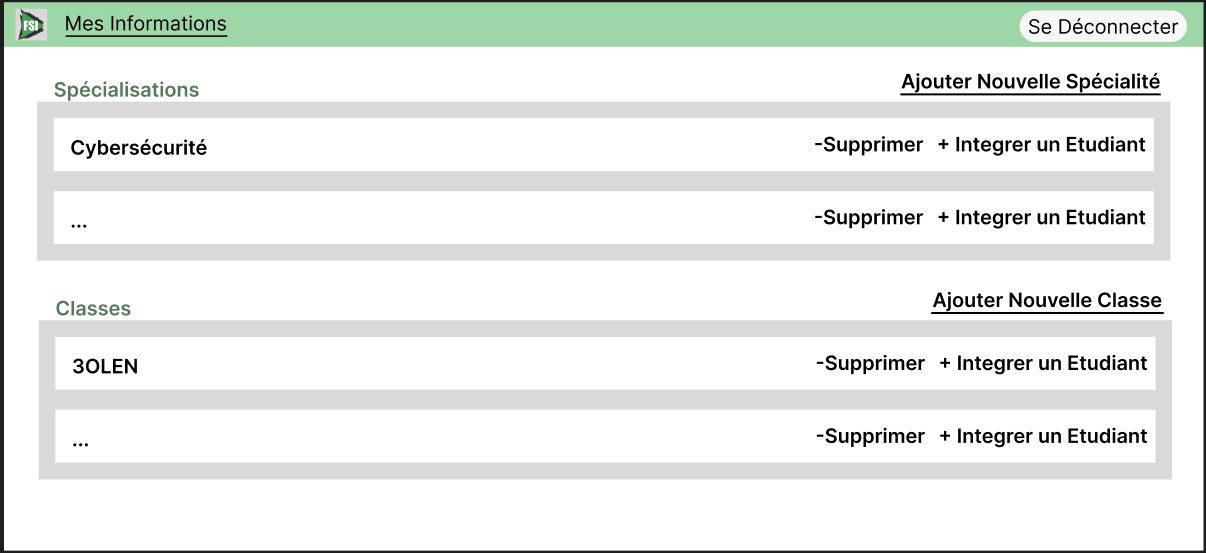
**Affectation Tuteurs**



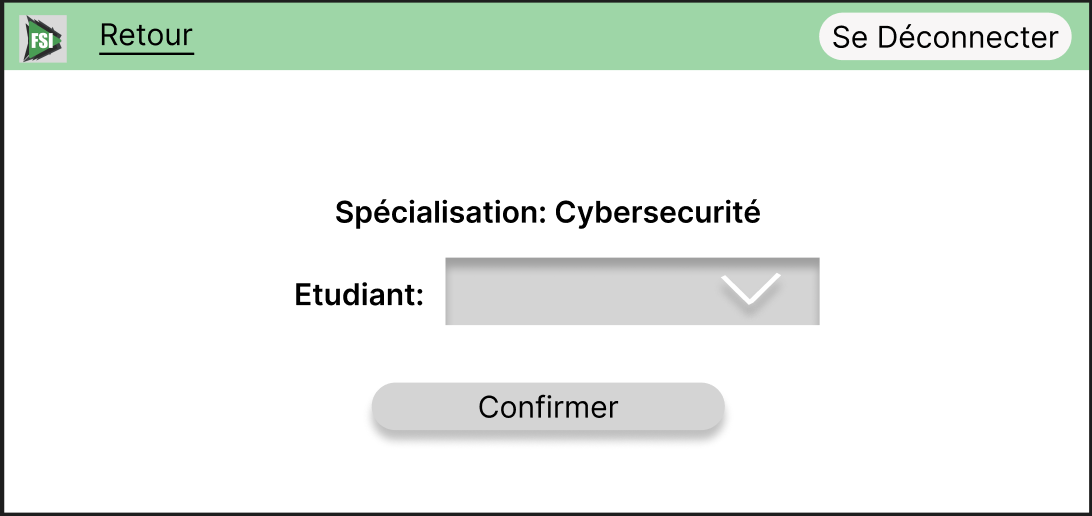
**Assignation Elève**



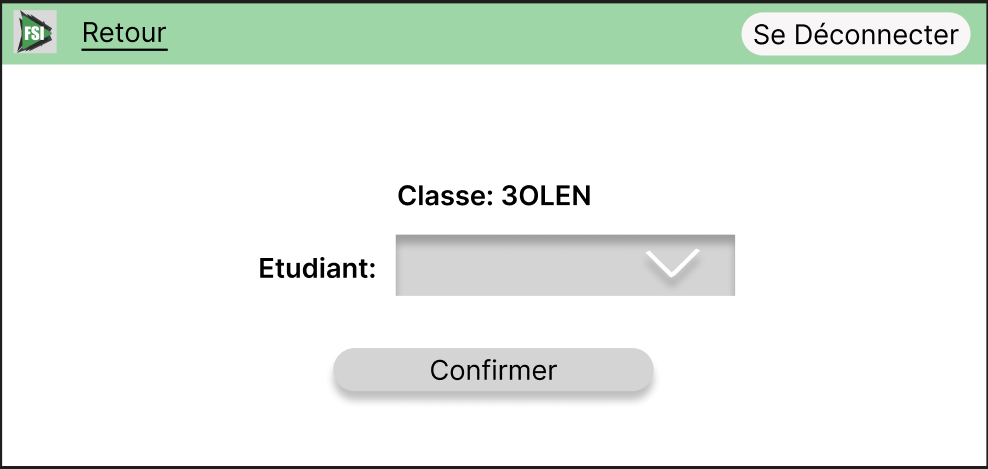
**Paramètres Généraux**



**Intégration Etudiant à Spécialisation**



**Intégration Etudiant à Classe**



# Base de données

## Dictionnaire de données

Le dictionnaire de données est un tableau qui liste toutes les informations que votre application va gérer. Chaque donnée correspond à une colonne dans une table de la base de données. Dans ce dictionnaire de données il est indiqué le nom de la donnée, sa description, son type, sa taille et si c’est une donnée calculée ou élémentaire.

Nous avons soigneusement analysé votre cahier des charges et les besoins que vous avez exprimés afin de nous assurer que toutes les informations nécessaires soient bien présentes.   
  
[Voir le dictionnaire de données](#_VI_Annexes)

## Dépendances fonctionnelles

Les **dépendances fonctionnelles** sont des règles qui montrent comment certaines informations dépendent les unes des autres dans une base de données. Elles permettent d'identifier quelles informations sont nécessaires pour retrouver et regrouper un ensemble de données.

En utilisant notre dictionnaire de données, nous avons pu identifier les dépendances fonctionnelles qui nous aident à organiser les informations dans les tables. Chaque dépendance fonctionnelle relie un identifiant unique comme un id à un ensemble d’informations.

Voici les dépendances fonctionnelles pour chaque entité de notre base de données :

**Entité : Utilisateur**

idUti 🡪 nomUti, preUti, mailUti, telUti, adrUti, cpUti, vilUti, altUti, logUti, mdpUti, idCla, idSpe, idTypUser, idEnt, idMai

**Entité : Maître Apprentissage**

idMai 🡪 nomMai, preMai, telMai, mailMai, idEnt

**Entité : Entreprise**

idEnt 🡪 nomEnt, adrEnt, cpEnt, vilEnt

**Entité : Classe**

idCla 🡪 nomCla, maxEtuCla

**Entité : Type Utilisateur**

idTypUser 🡪 typUser

**Entité : Spécialité**

idSpe 🡪 nomSpe

**Entité : Bilan1**

idBil1 🡪 notEnt1, notDos1, notOral1, moyBil1, rema1, idUti

**Entité : Bilan2**

idBil2 🡪 notDos2, notOral2, moyBil2, rema2, sujMem, idUti

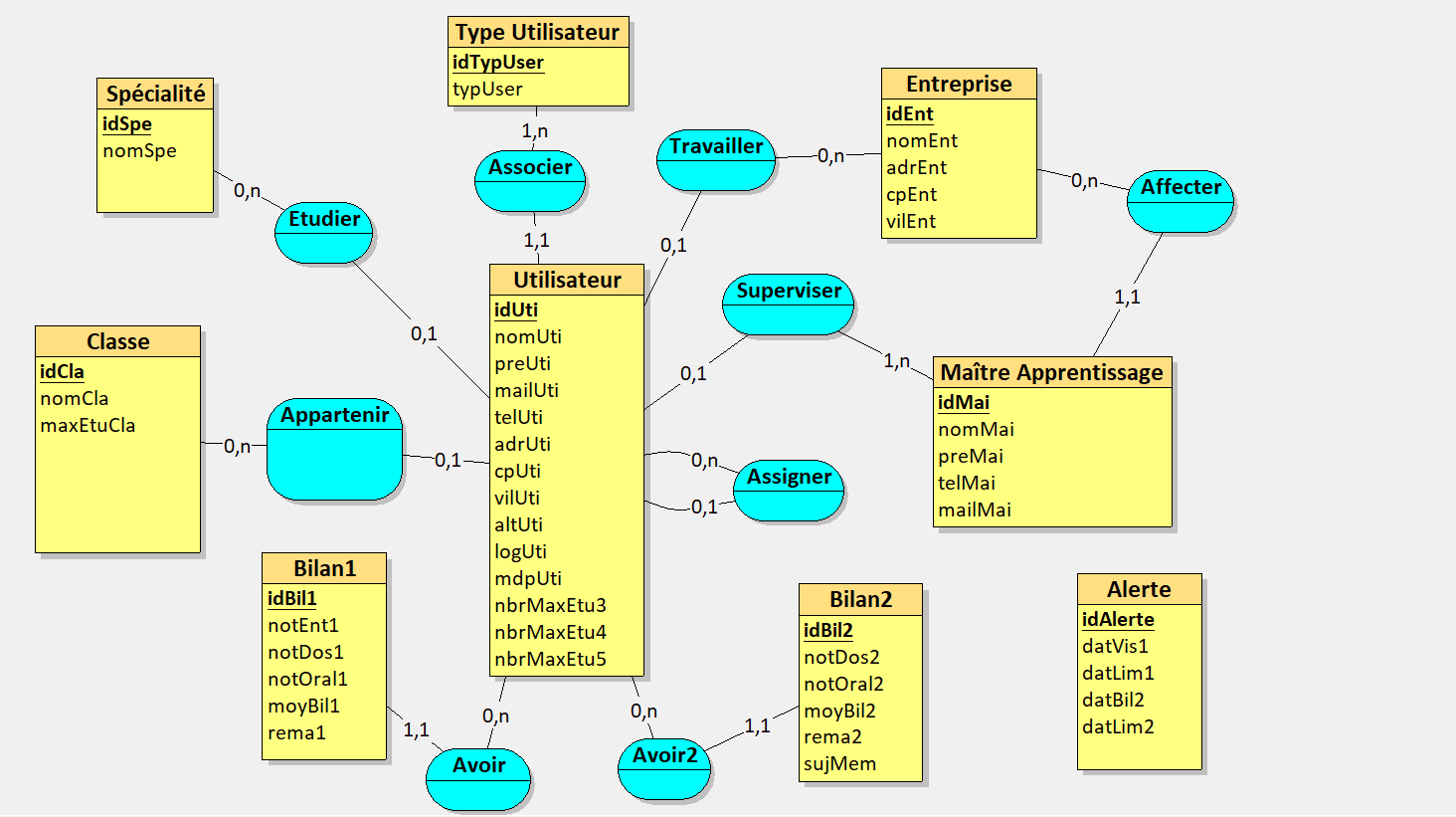
**Entité : Alerte**

idAlerte 🡪 datVis1, datLim1, datBil2, datLim2

## Modèle Conceptuel de Données (MCD)

Le Modèle Conceptuel de Données (MCD) est un schéma de la base de données qui représente l’organisation des données pour l’application.  
C’est en quelque sorte un plan qui montre l’organisation et les connexions des données.  
Son objectif est donc de structurer les données de manière logique et organisée afin de bien comprendre le fonctionnement avant de créer la base de données.

Par exemple la table **Utilisateur** contient toutes les informations nécessaires sur les utilisateurs de l'application. Chaque enregistrement dans cette table représente un utilisateur unique, qui est identifié par un identifiant (**idUti**). Cette table est au centre du MCD, car elle est liée à plusieurs autres entités, ce qui montre donc le rôle essentiel des utilisateurs dans l'application.



*Mcd utilisé pour le site web*

**Description des colonnes principales :**

**idUti :** Identifiant de l'utilisateur. C'est la clé primaire de la table, ce qui signifie qu'il ne peut pas y avoir deux utilisateurs avec le même identifiant dans la base de données.  
**nomUti :** Nom de l'utilisateur.  
…

**Relation avec d’autres tables :**

**Entreprise** : Les utilisateurs peuvent être associés à une entreprise pour indiquer leur employeur ou leur lieu de stage, grâce à la la clé primaire idEnt.  
…

## Structure des tables

La structure des tables quant à elle définit l’organisation et les caractéristiques des données dans chaque table de la base de données.  
Elle est essentielle pour garantir que toutes les informations nécessaires soient bien stockées, organisées et facilement accessibles pour les fonctionnalités de l’application.

Pour chaque table on peut retrouver le nom de la colonne, le type de donnée, sa taille, quels sont les clés primaires, si les champs peuvent êtres vides et la description de la donnée.

Table Utilisateur :

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

Description générée automatiquement

Table Maître d’Apprentissage :

Une image contenant texte, Police, nombre, ligne

Description générée automatiquement

Table Entreprise :

Une image contenant texte, Police, nombre, ligne

Description générée automatiquement

Table Classe :

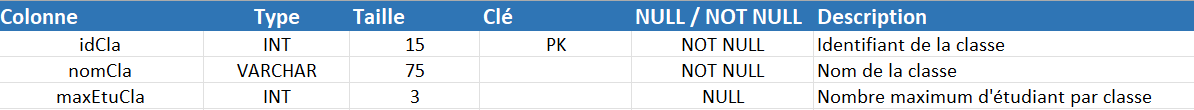


Table Type Utilisateur :

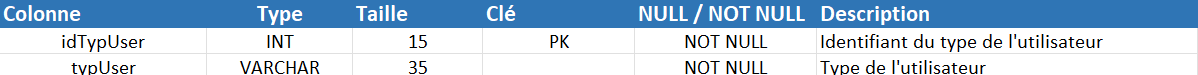


Table Spécialité :

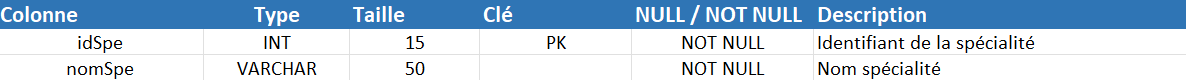


Table Bilan1 :

Une image contenant texte, Police, nombre, ligne

Description générée automatiquement

Table Bilan2 :

Une image contenant texte, Police, nombre, ligne

Description générée automatiquement

Table Alerte :

Une image contenant texte, Police, nombre, ligne

Description générée automatiquement

# Technique

Dans le cadre de ce projet, nous utiliserons le langage PHP natif ainsi que SQL sur MySQL afin de répondre aux exigences du client.

Pour le développement en PHP, nous travaillerons avec l’éditeur PHP Storm et utiliserons la version 8.2.0 de PHP pour assurer l’uniformité au sein de l’équipe.

Le projet intègrera également du HTML et du CSS pour gérer l’interface visuelle du site web.

PHP sera utilisé pour la gestion des données, notamment pour l’inscription et la connexion des utilisateurs à la base de données.

Nous utiliserons WAMP pour simuler le site web en local, ainsi que WhiteStarUML pour réaliser le diagramme de classes nécessaire à l’architecture du site.

Enfin, les maquettes du site web seront créées avec FIGMA, permettant ainsi de produire des maquettes professionnelles et d’assurer la cohérence visuelle du projet.

# Gestion projet

Le projet sera divisé en 5 lots de 2 semaines. Au cours de chaque lot, chaque membre de l’équipe se verra attribuer une ou plusieurs tâches afin de répartir le travail de manière équitable.

Voici la répartition des tâches pour chaque membre et pour chaque lot :

**Lot 1:**

* Chef de projet : Réaliser les Spécifications Technico-Fonctionnelles (STFD) du projet, mettre en place l'outil de gestion du projet avec Trello, configurer le système de gestion de GitLab et élaborer un planning détaillé du projet.
* Responsable BDD : Créer le dictionnaire de données qui décrit chaque information stockée dans la base de données et concevoir le Modèle Conceptuel des Données (MCD) pour représenter la structure de la base de données.
* Développeur Frontend : Réaliser les maquettes de tous les écrans de l'application à l'aide de l'outil FIGMA.
* Développeur Backend : Créer le diagramme de classe BO pour modéliser les objets métier de l'application.

**Lot 2:**

* Chef de projet : Créer le cahier de recette qui détaille les tests à réaliser et effectuer les tests du CRUD codé par le backend.
* Responsable BDD : Créer un jeu de données de test pour alimenter la base de données et alimenter la base de données avec les données de test.
* Développeur Frontend : Créer le CSS pour définir l'apparence et le style de l'application et développer la page d'accueil pour les tuteurs.
* Développeur Backend : Réaliser les BO qui encapsulent la logique métier de l'application et réaliser les CRUD (Create, Read, Update, Delete) des DAO pour gérer les interactions avec la base de données.

**Lot 3:**

* Chef de projet : Développer la page "Mes informations étudiants".
* Responsable BDD : Développer la page de connexion à l'application et mettre en place la gestion des sessions utilisateur.
* Développeur Frontend : Développer la page "Liste étudiants".
* Développeur Backend : Développer la page "Alertes".

**Lot 4:**

* Chef de projet : Développer les pages "Mes Bilans" et "Paramètres".
* Responsable BDD : Développer la page "Mes informations tuteurs".
* Développeur Frontend : Développer les pages "Bilan 1" et "Bilan 2".
* Développeur Backend : Développer la page "Détail d'un étudiant".

**Lot 5:**

* Chef de projet : Développer la page "Gestion Tuteur" et préparer la mise en production de l'application.
* Responsable BDD : Développer la page d'accueil pour les étudiants et l'administrateur et développer les pages "Mes Bilans" pour les étudiants.
* Développeur Frontend : Finaliser les paramètres généraux de l'interface utilisateur et vérifier la cohérence visuelle de l'application, en s'assurant que toutes les pages ont un style uniforme.
* Développeur Backend : Développer la page "Affectation Tuteur-Etudiant" et développer la page "Gestion Etudiant".

Le projet aura comme arborescence :

A screenshot of a computer program

Description automatically generated

Cette arborescence respect le MVC.

## VI.I Trello

Le Trello permet d’attribuer les tâches aux différents membres de l’équipe et de suivre l’avancement des tâches en temps réel.

Voici notre Trello, qui sera utilisé pour la gestion de ce projet :

Screens screenshot of a computer screen

Description automatically generated

Le projet est divisé en plusieurs lots, ce qui nous permettra de progresser étape par étape.

Chaque tâche est attribuée à différents développeurs au sein de l’équipe, permettant ainsi une avancée rapide et fluide.

## VI.II Planning

Lors de la réalisation de ce projet, nous travaillerons par lot qui dureront 2 semaines.

Le planning a été fait avec Gant Project qui nous permet de visualiser les tâches.

Voici la démo du planning :

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Planning détaillé du projet avec chaque lot et chaque tâche.

A white paper with black text

Description automatically generated

Planning détaillé du projet, avec la répartition des lots et des tâches.

# VI. Annexes

Voici le dictionnaire de données utilisé pour l’application web.

A screenshot of a computer

Description automatically generated

Branch du git :

A screenshot of a computer

Description automatically generated

Chaque membre à sa branche afin qu’il puisse déposer ces fichiers sans écraser ceux d’un autre.

Répertoire git :

A screenshot of a computer

Description automatically generated

Lieu où les documents seront déposés, ici nous avons les documents du Chef de projet.