Wissem Laffi 2SIO

Yanis Bouquet

Alexis Beriot

Saadia Zakariyya

***Spécifications***

***Fonctionnelles***

Sommaire

[I. Description du projet 3](#_Toc181865052)

[II. Modélisation UML 5](#_Toc181865053)

[III. Ecran 6](#_Toc181865054)

[IV. Base de données 10](#_Toc181865055)

[V. Technique 15](#_Toc181865056)

[VI. Gestion projet 16](#_Toc181865057)

[VI.I Trello 19](#_Toc181865058)

[VI.II Planning 21](#_Toc181865059)

# 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.

Le projet vise à développer une application web pouvant gérer le suivi des élèves ainsi que des tuteurs au seins d’un établissement scolaire.

Cette application web sera divisé en plusieurs parties :

* Une page inscription / connexion pour les élèves et les tuteurs
  + Pour les tuteurs, permettre l’ajout de note, 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 tels que la spécialité, la classe et le nom et prénom des tuteurs

L’application web aura comme nom : e-fsi

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 étudiant.

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 site doit être fait en SQL sous mySQL et en php
* Chaque front doit être cohérant 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, …).

# Modélisation UML

Pour ce projet, une modélisation en amont doit être effectuer avec les liens entre les classes. Voici le diagramme UML que nous utiliserons pour ce site web. Pour la modélisation UML, j'ai décidé de faire ça sur whitestaruml car c'est un logiciel qui m'est familier pour faire des diagrammes de classe.

Dans mon diagramme de classe, j'ai décidé de mettre les classes Bilan1 et Bilan2 qui héritent d'un classe Bilan. Cet héritage me paraissait évident car mise à part le sujet de mémoire, tous les autres attributs doivent être présent dans les deux classes.

De la même manière, j'ai fait hériter mes classes administrateur, tuteur et étudiant à une classe utilisateur. Pour le reste j'ai fait en fonction de ma base de données.

A screenshot of a computer screen

Description automatically generated

# Ecran

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

**Page de Connexion**

Page que les utilisateurs auront pour s’identifierA screenshot of a login form

Description automatically generated

**Page d’Accueil (Etudiant)**

**Page d’accueil pour les étudiants**

A screenshot of a computer

Description automatically generated

**Page Mes Infos (Etudiant)**

Page d’information d’un étudiant

## A screenshot of a phone Description automatically generated

**Page Bilan**

Bilan des notes d’un étudiant

## A screenshot of a computer Description automatically generated

**Page d’Accueil (Tuteur)**

Page d’accueil pour le tuteur

## 

**Page Mes Infos (Tuteurs)**

Page d’information d’un tuteur

## A screenshot of a computer Description automatically generated

## 

## 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

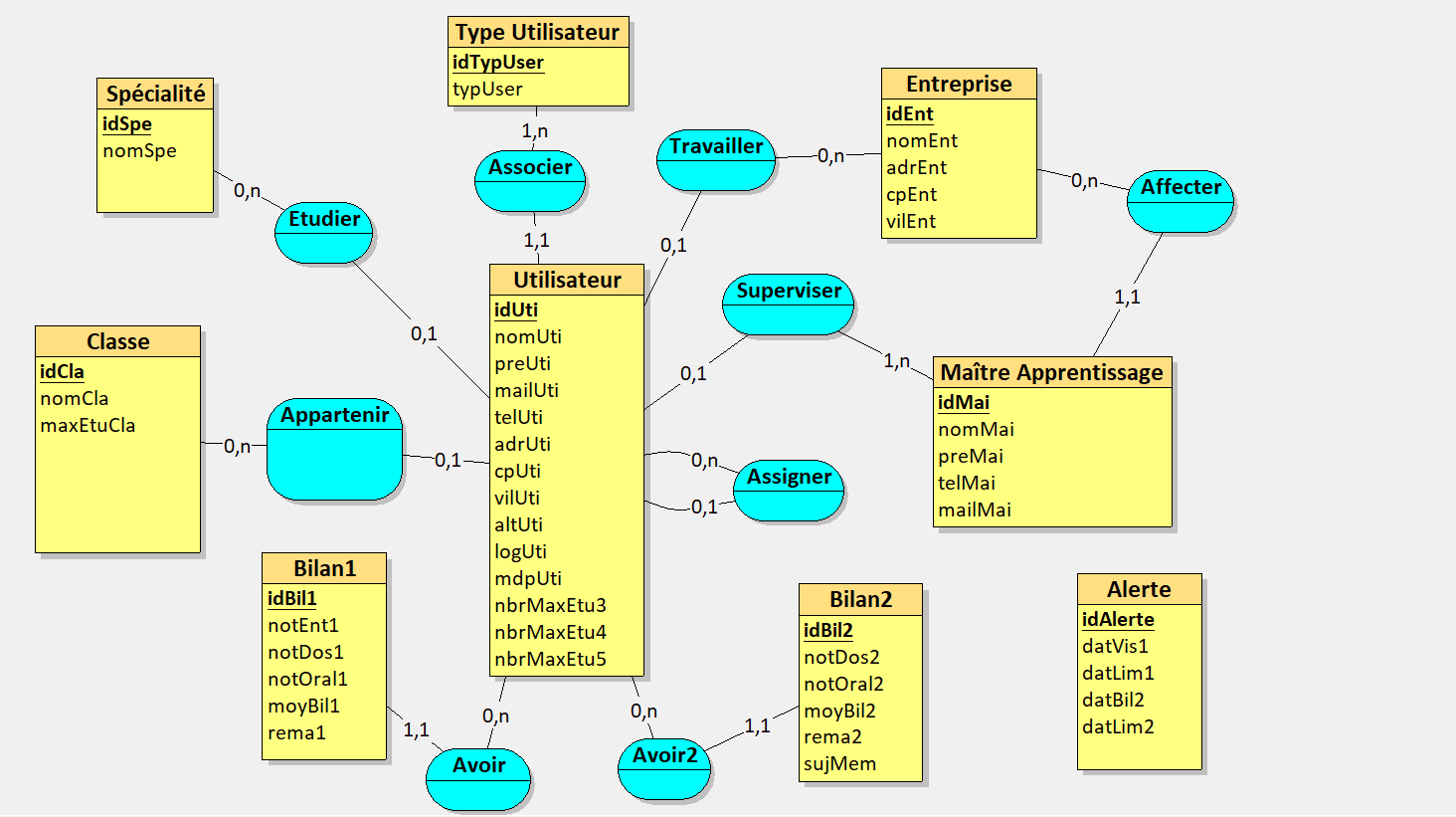
**Entité : Alerte**

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 :

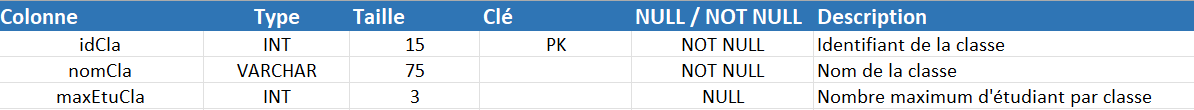


Table Type Utilisateur :

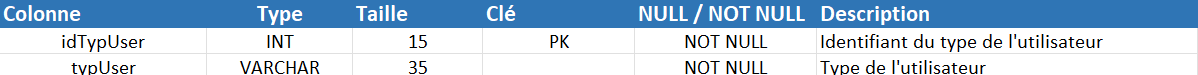


Table Spécialité :

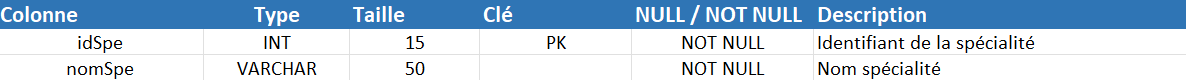


Table Bilan1 :

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, capture d’écran, Police, nombre Description générée automatiquement

# Technique

Lors de ce projet nous utiliserons du PHP native ainsi que SQL sous MySQL afin de pouvoir respecté les conditions du client.

Pour le php, nous utiliserons le logiciel PhpStorm et la version du php sera le 8.2.0 pour l’ensemble de l’équipe.

Il y aura également du HTML/CSS pour pouvoir gérer la partie visuelle du site web.

Le php nous permettra de gérer les données avec la base de données, l’inscription et la connexion.

Nous utiliserons wamp pour pouvoir simuler le site web en local et nous utiliserons LucidChart afin de faire le diagramme de classe que nous utiliserons pour le site web.

Pour les maquettes du site web, nous utiliserons FIGMA qui nous permettra de faire des maquettes professionnelles.

# Gestion projet

Le projet sera divisé en 5 lots de 2 semaine. Durant ces lots, chacun des membres de l’équipe aura une ou plusieurs tâches afin de répartir en travail équitable.

Voici les tâches, pour chaque membre et chaque lot :

**Chef de projet**

* **Lot 1:**
  + 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.
  + Élaborer un planning détaillé du projet.
* **Lot 2:**
  + Créer le cahier de recette qui détaille les tests à réaliser.
  + Effectuer les tests du CRUD codé par le backend .
* **Lot 3:**
  + Développer la page "Mes informations étudiants".
* **Lot 4:**
  + Développer les pages "Mes Bilans" et "Paramètres".
* **Lot 5:**
  + Développer la page "Gestion Tuteur".
  + Préparer la mise en production de l'application.

**Responsable BDD**

* **Lot 1:**
  + Créer le dictionnaire de données qui décrit chaque information stockée dans la base de données.
  + Concevoir le Modèle Conceptuel des Données (MCD) pour représenter la structure de la base de données.
* **Lot 2:**
  + Créer un jeu de données de test pour alimenter la base de données.
  + Alimenter la base de données avec les données de test.
* **Lot 3:**
  + Développer la page de connexion à l'application.
  + Mettre en place la gestion des sessions utilisateur.
* **Lot 4:**
  + Développer la page "Mes informations tuteurs".
* **Lot 5:**
  + Développer la page d'accueil pour les étudiants et l'administrateur.
  + Développer les pages "Mes Bilans" pour les étudiants.

**Développeur Frontend**

* **Lot 1:**
  + Réaliser les maquettes de tous les écrans de l'application à l'aide de l'outil FIGMA.
* **Lot 2:**
  + Créer le CSS pour définir l'apparence et le style de l'application.
  + Développer la page d'accueil pour les tuteurs.
* **Lot 3:**
  + Développer la page "Liste étudiants".
* **Lot 4:**
  + Développer les pages "Bilan 1" et "Bilan 2".
* **Lot 5:**
  + Finaliser les paramètres généraux de l'interface utilisateur.
  + Vérifier la cohérence visuelle de l'application, en s'assurant que toutes les pages ont un style uniforme.

**Développeur Backend**

* **Lot 1:**
  + Créer le diagramme de classe BO pour modéliser les objets métier de l'application.
  + Créer le diagramme de classe des DAO pour modéliser l'accès aux données.
* **Lot 2:**
  + Réaliser les BO qui encapsulent la logique métier de l'application.
  + Réaliser les CRUD (Create, Read, Update, Delete) des DAO pour gérer les interactions avec la base de données.
* **Lot 3:**
  + Développer la page "Alertes".
* **Lot 4:**
  + Développer la page "Détail d'un étudiant".
* **Lot 5:**
  + Développer la page "Affectation Tuteur-Etudiant".
  + Développer la page "Gestion Etudiant".

Le projet aura comme arborescence :

A screenshot of a computer

Description automatically generated

Cette arborescence respect le MVC.

## VI.I Trello

Le trello permet d’affecter les tâches au différent membre de l’équipe. On suit également les avancements des tâches en temps réel.

Voici notre trello, qui sera utilisé pour ce projet.

Screens screenshot of a computer screen

Description automatically generated

Le projet est divisé en plusieurs lot qui nous permettrons d’avancer étape par étape.

Chaque tâche est attribuée à différents programmeur au sein de l’équipe, permettant d’avancer de manière rapide et fluide.

## 

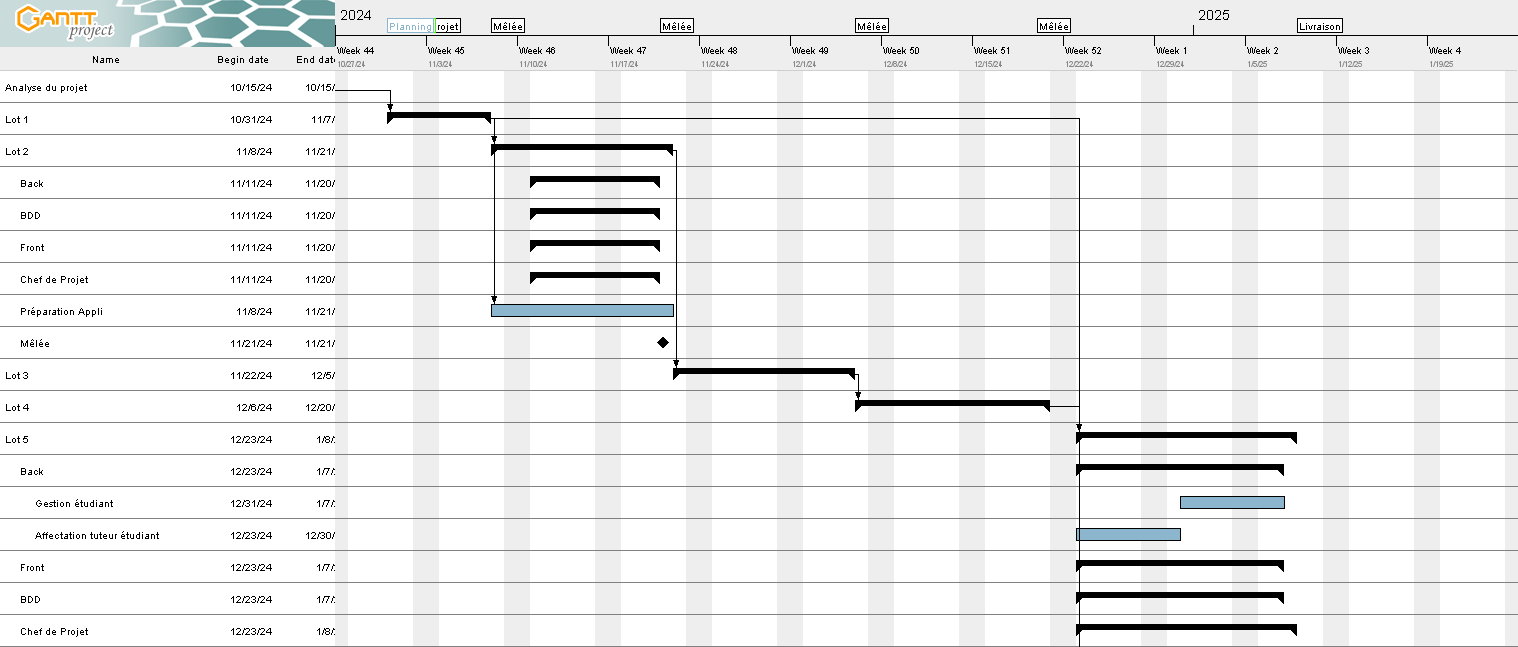
## VI.II Planning

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

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

Voici le demo du planning :

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 chaque lot et chaque tâche.

# 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 déposé ces fichiers sans écraser ceux d’un autre.

Répertoire git :

A screenshot of a computer

Description automatically generated

Lieu où les documents seront déposer, ceci sont les documents du Chef de projet.