

*-∞ Unlimited Studies ∞-*

# **Système d'apprentissage assisté**

Dossier de conception technique

Version 1.0  
06/2021

**Buglioni Thomas**

<b>I. Présentation générale.....</b>	<b>1</b>
1. Contexte .....	1
2. Besoins à l'origine du projet .....	1
3. Idée à l'origine du projet .....	1
<b>II. Diagramme de classe.....</b>	<b>2</b>
1. Graphique.....	2
2. Description du graphique .....	2
<b>III. Modele physique de données .....</b>	<b>4</b>
1. Graphique.....	4
2. Description du graphique .....	4
<b>IV. Diagramme de composants .....</b>	<b>5</b>
1. Graphique.....	5
2. Description du graphique .....	5
<b>V. Diagramme de composants .....</b>	<b>7</b>
1. Graphique.....	7
2. Description du graphique .....	7

# I. Présentation générale

---

## 1. Contexte

« Unlimited Studies » est un projet personnel dans le cadre de mon diplôme de développeur python en 2021.

Il est orienté sur l'usage de python, avec django, une approche fullstack (front/back-end) et enfin une approche OPS pour le déploiement (docker/ubuntu/digital-ocean).

---

## 2. Besoins à l'origine du projet

Ce projet est basé sur une problématique simple :

Comment apprendre un element (un savoir, une langue, quelque chose en mémoire) de manière optimisé ?

Donc il est question de compréhension de l'element, sa rétention, et enfin sa révision.

La dimension psychologique autour de la « charge cognitive » est fondamentale et à la base de ce projet

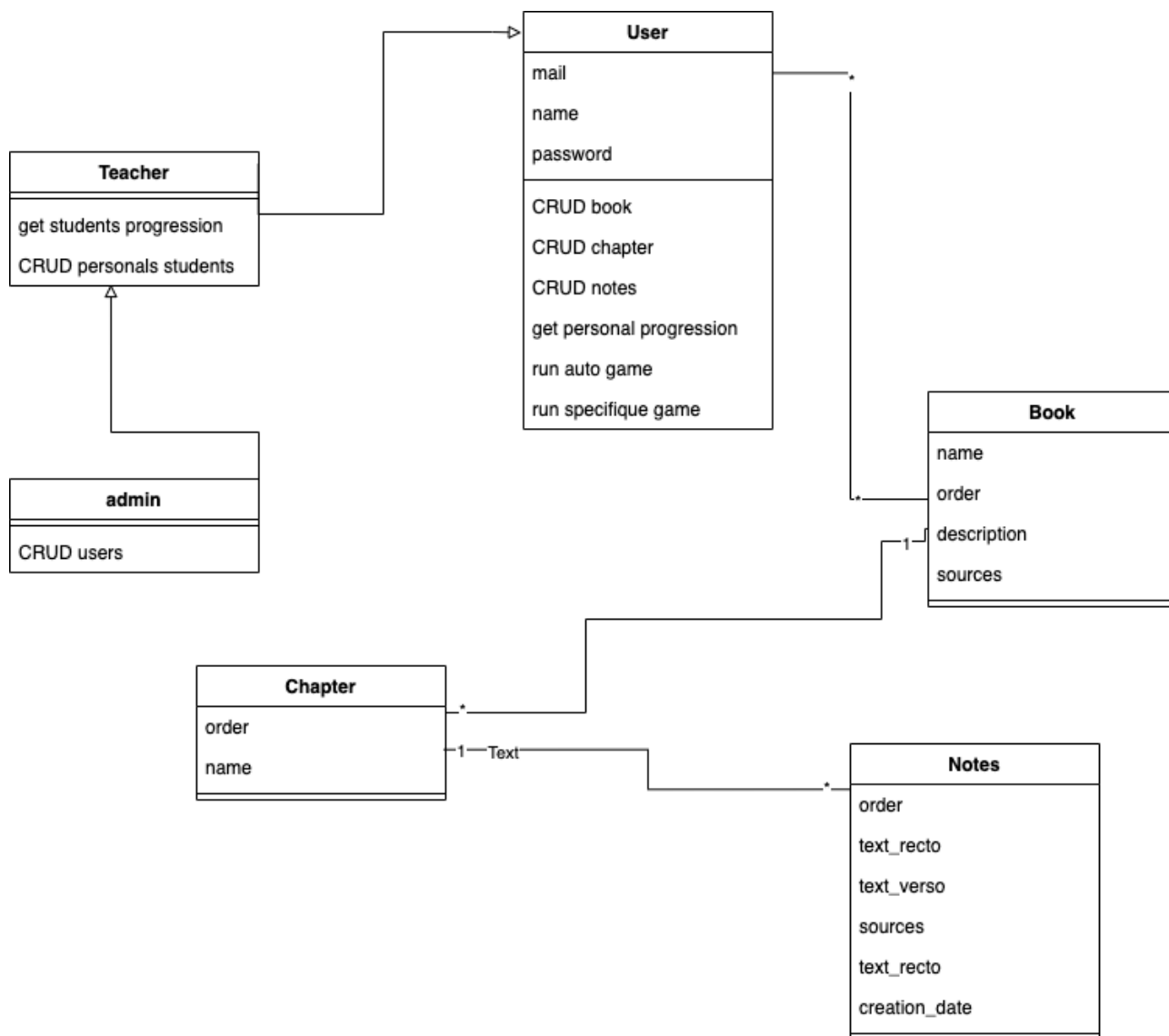
---

## 3. Idée à l'origine du projet

- avoir un logiciel en ligne
- Pouvoir se connecter et avoir ses « notes » à apprendre
- Avoir une revision espacé dans le temps selon

## II. Diagramme de classe

### 1. Graphique



### 2. Description du graphique

Chaque utilisateur possède 1 ou plusieurs « book »

Chaque « book » possède 1 ou plusieurs « chapter »

Chaque « chapter » possède 1 ou plusieurs « notes »

Exemple :

Louis a le livre « anglais », avec le chapitre « vocabulaire de base » et dedans « hello=bonjour; one=1; two=2 »

- Chaque utilisateur a une progression sur l'étude des notes
- Chaque utilisateur peut voir sa progression
- Chaque utilisateur peut étudier de manière automatique les notes ou de manière spécifique (un chapitre)

Un professeur a les mêmes fonctions que un utilisateur mais peut également inviter d'autres utilisateurs à étudier ses « books »

Chaque professeur peut suivre la progression de ses élèves dans un book qui le concerne.

Exemple

- John a le livre mathématique de 3ème (il est professeur) en écriture/étude/lecture
- Hervé a le livre mathématique de 3ème de John en étude/lecture uniquement
- Laura a le livre mathématique de 3ème de John en étude/lecture uniquement

John peut voir la progression de Hervé et de Laura

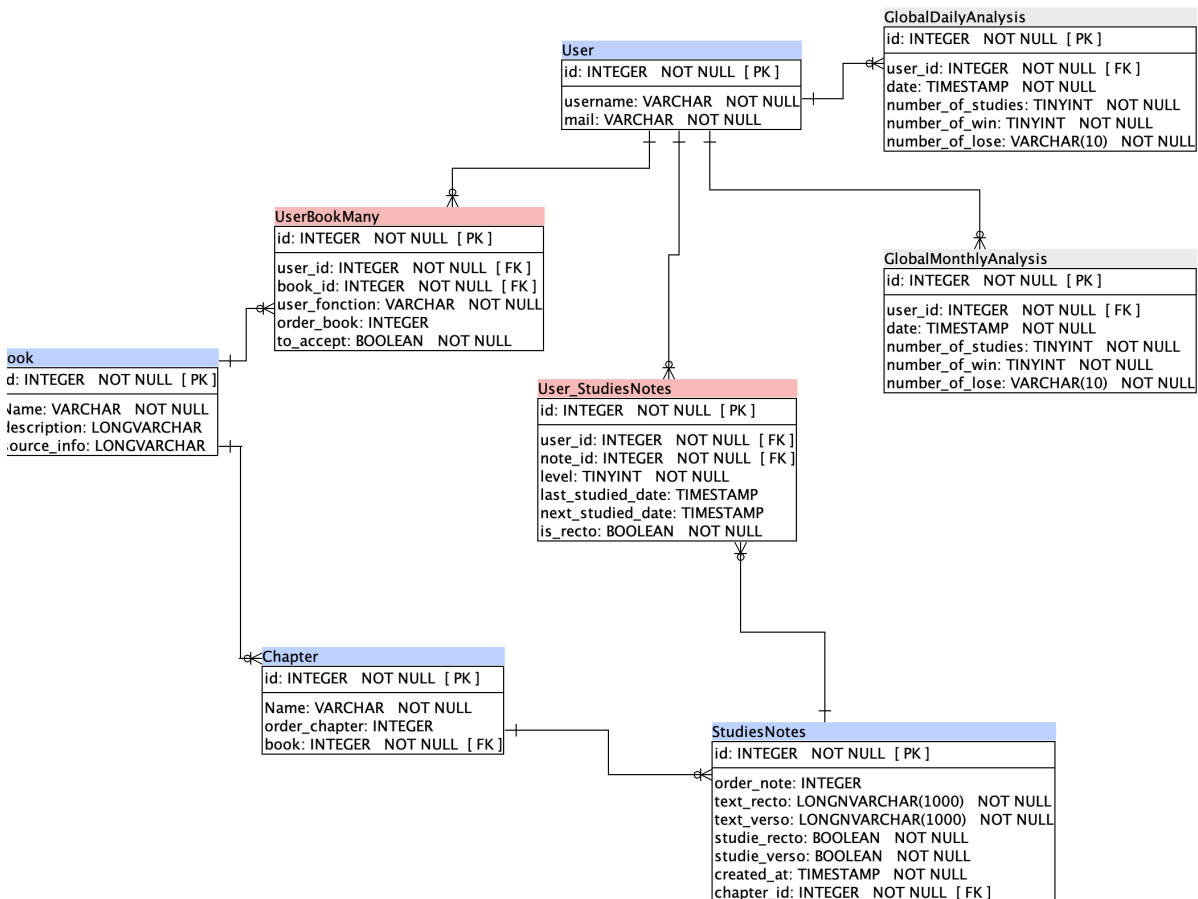
Hervé ne peut pas voir la progression de Laura et inversement

Les élèves n'ont pas accès à la progression du professeur

Un administrateur a les mêmes fonctions que un utilisateur et un professeur mais peut également gérer les utilisateurs et faire des modifications sur la bibliothèque officielle de livre (fonctions étendu de « user »)

# III. Modele physique de données

## 1. Graphique



## 2. Description du graphique

Section bleue (Donnée élémentaire) :

- User : qui ?
- Book/chapter/studiesNotes : quoi ?

Section rouge (tables associations)

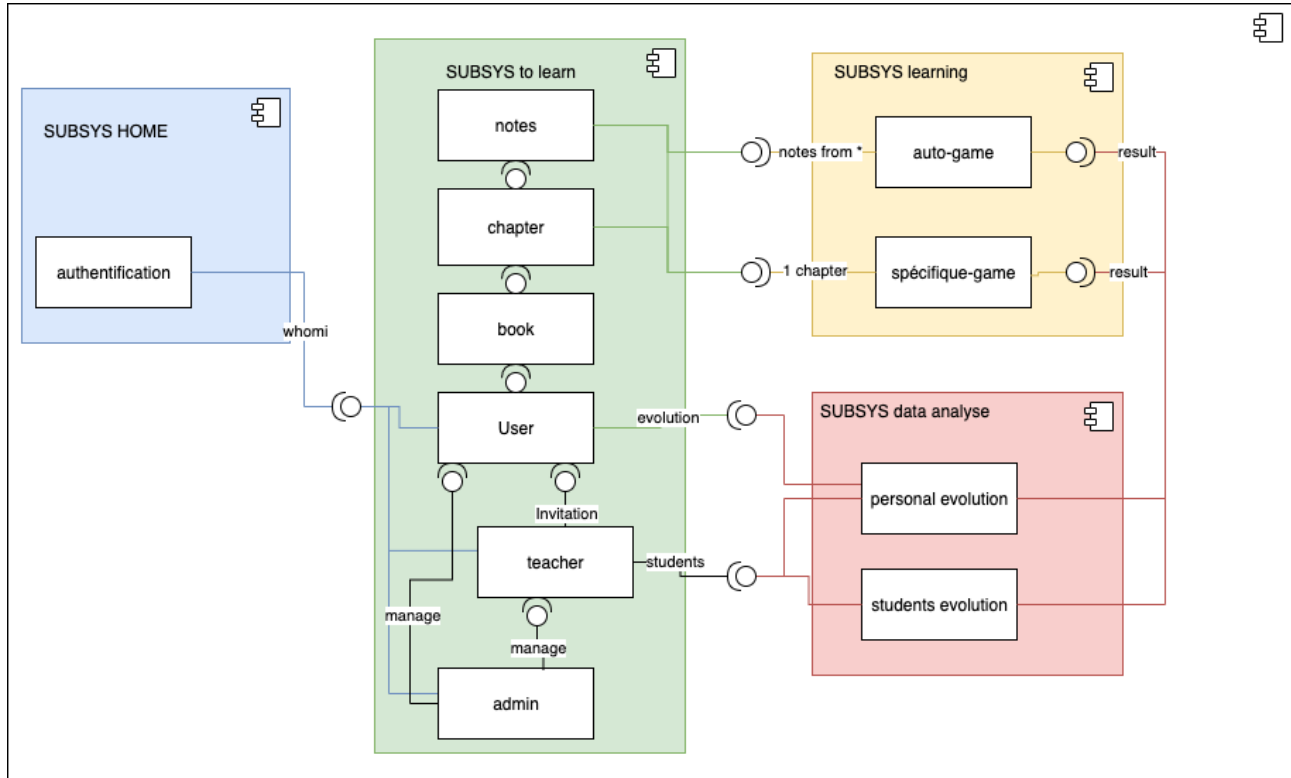
- userBookMany: a qui appartient les données élémentaires
- User\_StudiesNotes: quelle progression pour un utilisateur par studiesNotes

Section grise (donnée concernant la progression)

- Progression sur les 10 derniers jours et 12 derniers mois

## IV. Diagramme de composants

### 1. Graphique



### 2. Description du graphique

4 sections

HOME.

- information
- Connection

LEARN.

- creation de support à apprendre par user : book>chapters>notes
- Hierarchie : professeur>eleves
- Gestion générale : admin

LEARNING.

- étude automatisé des notes
- etude d'un chapitre spécifique

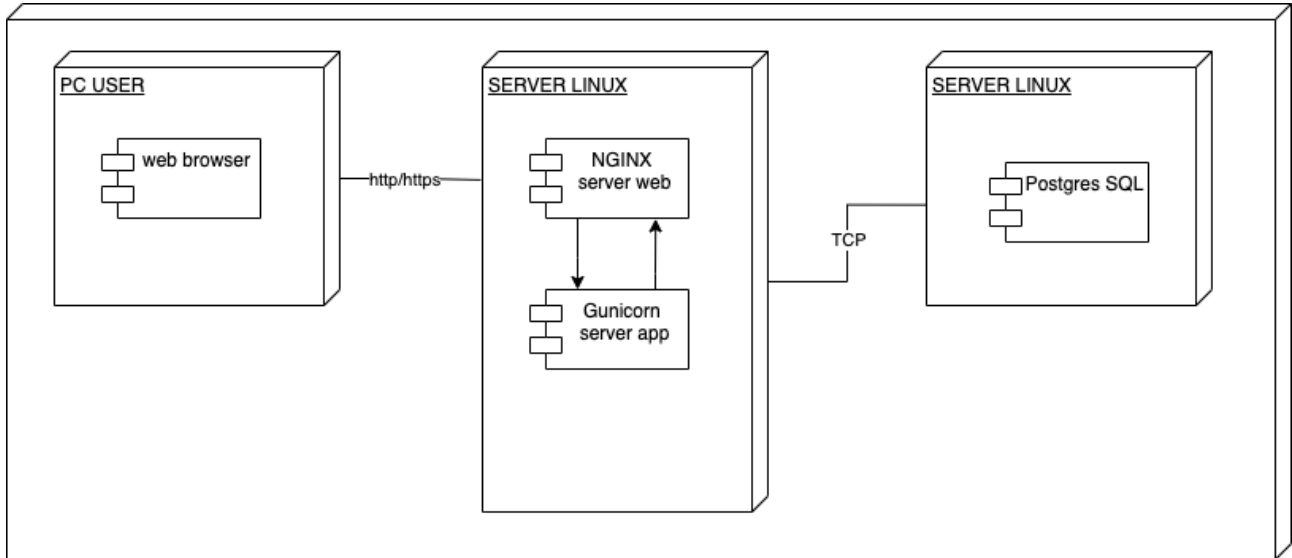
Evolution.

- évolution personnelle
- Evolution d'un ou plusieurs élèves



## V. Diagramme de composants

### 1. Graphique



### 2. Description du graphique

Le système « pc user » accède au server principal (linux) via HTTP/HTTPS.

Le serveur principal gère l'application via le serveur web Nginx et le server app Gunicorn qui communiquent entre eux.

Le serveur principal communique avec le serveur secondaire (linux) afin de gérer la base de donnée mysql via TCP