

# Projet 13 - Projet final : prêt pour le feu d'artifices ?

Parcours OpenClassrooms - Développeur d'application Python

Étudiant : Guillaume OJARDIAS      Mentor : Erwan KERIBIN

Mentor évaluateur : Jimmy Kuassi KUMAKO

# Sommaire

<b>I. Présentation</b>	<b>3</b>
I.1. Liens du projets . . . . .	3
I.2. Contexte du projet . . . . .	3
I.3 Organisation du projet . . . . .	3
<b>II. Démarche de création</b>	<b>3</b>
II.1 Stack technique . . . . .	3
II.2. Modèle de données . . . . .	4
II.3. Tests de l'application . . . . .	4
<b>III. Bilan</b>	<b>4</b>
III.1. Prise en main de la stack . . . . .	4
III.2. Planification du projet . . . . .	4
III.3. Priorisation des tâches . . . . .	4

# I. Présentation

## I.1. Liens du projets

- Le code source du projet est disponible sur la plate-forme GitHub à cette adresse : <https://github.com/GuillaumeOj/P13-WOD-Board>.
- Le site est visible en ligne à cette adresse : <http://projet-13.ojardias.io/>.
- La roadmap est accessible ici : <https://www.notion.so/guillaumeoj/>.

## I.2. Contexte du projet

Il s'agit du projet final de la formation développeur d'application Python, proposé par OpenClassrooms. Le but de cette application est de permettre aux pratiquants du Crossfit® de tenir un journal de bord de leurs séances. La note d'intention (voir *P13\_01\_Note\_intention*) détail les objectifs de ce projet.

## I.3 Organisation du projet

Pour planifier et organiser le projet, une roadmap sous forme de Kanban à était réalisée. La roadmap est disponible ici : <https://www.notion.so/guillaumeoj/>. Elle se découpe en quatre colonnes:

1. To Do : tâches à réaliser;
2. In Progress : tâches en cours de réalisation;
3. Done : tâches terminées;
4. Backlog : tâches prévues.

Pour la planification, chaque tâche s'est vue attribuer une date de début et de fin. Ces dates ont permis d'estimer le temps nécessaire à la réalisation d'un Minimum Viable Product.

# II. Démarche de création

## II.1 Stack technique

L'application est divisée en deux parties distinctes : back-end et front-end.

La partie back-end repose sur l'utilisation du framework FastAPI (<https://fastapi.tiangolo.com/>). La base de données est gérée avec PostgreSQL (<https://www.postgresql.org/>). L'interface avec le langage Python se fait grâce à l'ORM SQLAlchemy (<https://www.sqlalchemy.org/>).

La partie front-end est une application ReactJS (<https://reactjs.org/>). Cette application utilise l'API créée grâce à FastAPI pour communiquer avec le back-end.

Ce choix va à l'encontre des technologies rencontrées lors de notre parcours de formation. Il s'agissait de profiter de la formation pour découvrir ces frameworks avant l'entrée sur le monde du travail.

## II.2. Modèle de données

L'application minimale contient 6 grandes classes :

1. **User** : pour les informations relatives à l'utilisateur de l'application;
2. **Wod** : correspond à un exercice (Workout Of the Day);
3. **Round** : est un tour dans un Wod, un Wod peut être composé de un ou plusieurs tours;
4. **Movement** : décrit un mouvement de base en Crossfit® (pompe, traction, etc.);
5. **Goal** : détaille un Movement avec un nombre de répétition et ou un temps imposé pour sa réalisation;
6. **Equipment** : liste les équipements utilisable au Crossfit® (dumbbell, barre de traction, etc.).

Les 2 autres classes sont les suivantes :

1. **WodType** : définit la catégorie de Wod, nécessaire au calcul du score final;
2. **UnitType** : est une classe générique pour lister les unités de mesures (mètres, kilogrammes, etc.).

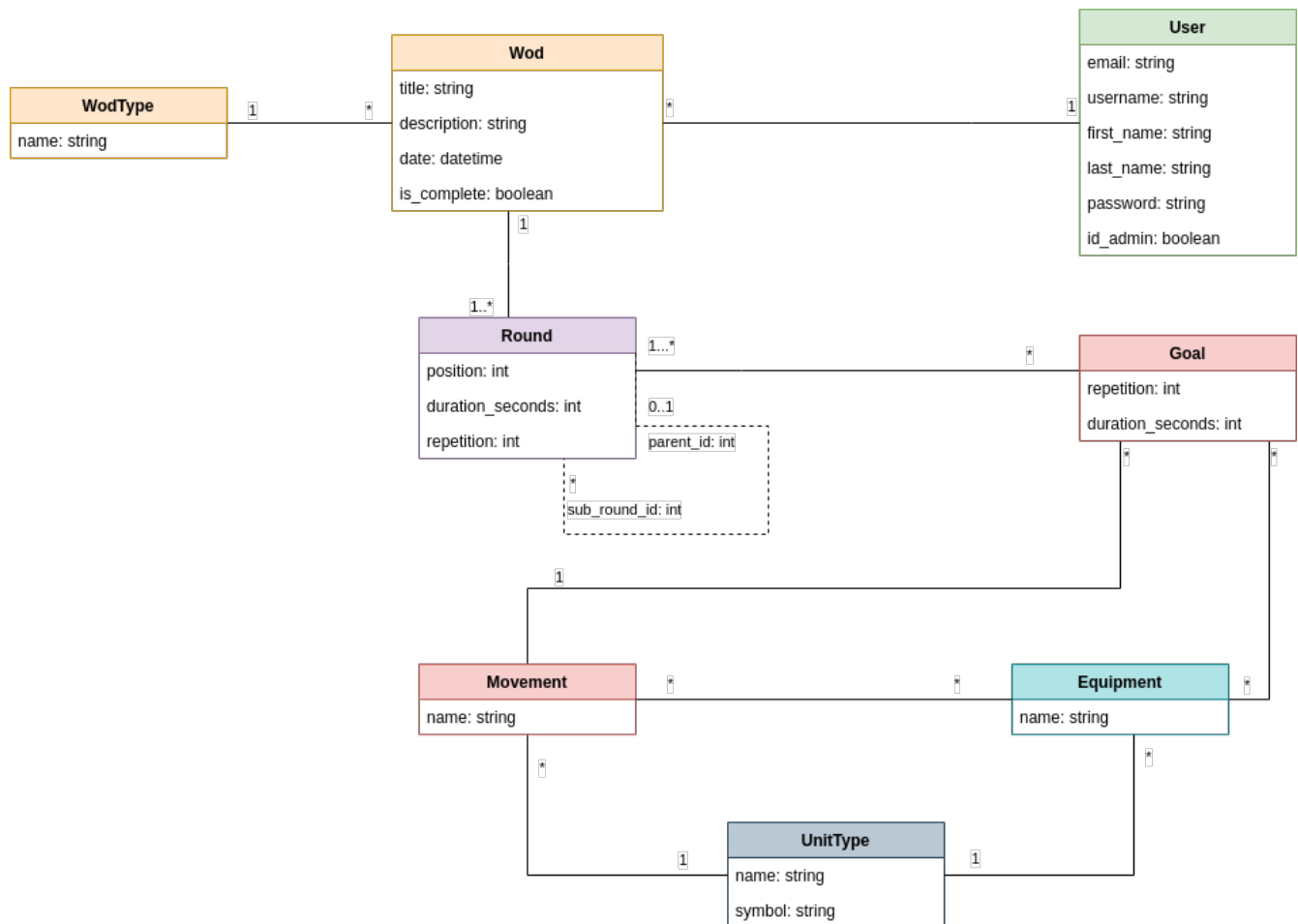


Figure 1: Diagramme de classes

## II.3. Tests de l'application

## III. Bilan

### III.1. Prise en main de la stack

### III.2. Planification du projet

### III.3. Priorisation des tâches