

Projet 13 - Rapport

Parcours OpenClassrooms - Développeur d'application Python

Étudiant : Guillaume OJARDIAS Mentor : Erwan KERIBIN
Mentor évaluateur : Jimmy Kuassi KUMAKO

Sommaire

I. Présentation	2
I.1. Liens du projets	2
I.2. Contexte du projet	2
I.3 Organisation du projet	2
II. Démarche de création	2
II.1 Stack technique	2
II.2. Modèle de données	3
II.3. Tests de l'application	3
III. Bilan	4
III.1. Prise en main de la stack technique	4
a. Front-end	4
b. Back-end	4
III.2. Planification du projet	5
III.3. Fonctionnalités	5
III.4 Bilan de projet	6

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/>.
- L'ensemble des livrables sont disponibles ici : <https://github.com/GuillaumeOj/P13-WOD-Board/tree/main/report>

I.2. Contexte du projet

Il s'agit du projet final de la formation développeur d'application Python, proposée 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 a été 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 base de données / application se fait grâce à l'ORM SQLAlchemy (<https://www.sqlalchemy.org/>).

Le front-end est une application ReactJS (<https://reactjs.org/>). Cette application utilise l'API du back-end pour communiquer avec ce dernier.

Ces technologies sont légèrement différentes de celles rencontrées lors de notre parcours de formation. Je souhaitais profiter du statuts de projet "scolaire" pour découvrir ces frameworks avant l'entrée sur le monde du travail.

II.2. Modèle de données

L'application contient 6 grandes classes :

1. **User** : pour les informations relatives à l'utilisateur;
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.), un **Round** peut avoir un ou plusieurs **Movement**;
5. **Goal** : est un **Movement** avec un nombre de répétitions et ou un temps imposé pour sa réalisation;
6. **Equipment** : liste les équipements utilisables au Crossfit® (dumbbell, barre de traction, etc.).

Les 2 autres classes sont les suivantes :

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

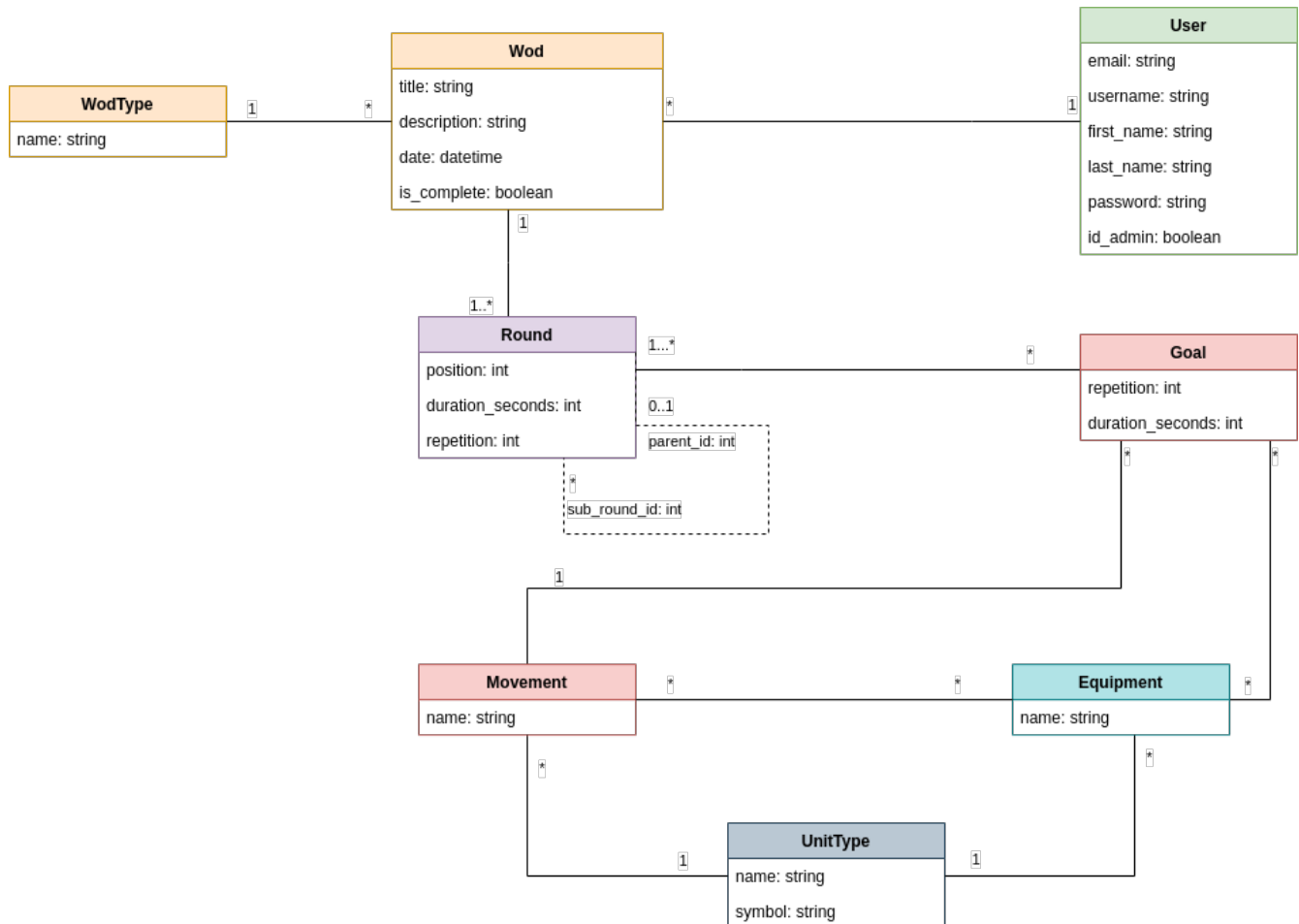


Figure 1: Diagramme de classes

II.3. Tests de l'application

Les tests sont réalisés sur la partie back-end de l'application. Les tests back-end couvrent 98% du code. Un accent particulier a été mis sur :

- les **routeurs** de l'API (points d'entrée);
- les fonctions de **CRUD** (Create Read Update Delete).

Ces tests ont été réalisés dans l'esprit du TDD. Ils sont visibles ici : <https://github.com/GuillaumeOj/P13-WOD->

Board/tree/main/backend/wod_board/tests

Les tests de l'application front-end ont été mis de côté, par manque de temps et de connaissance du framework de test de ReactJS. Ils feront l'objet d'un travail de refactorisation à la suite de la soutenance.

III. Bilan

III.1. Prise en main de la stack technique

a. Front-end

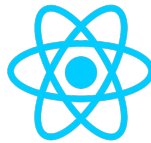


Figure 2: ReactJS

Le front-end est un point que je souhaite retravailler fortement par la suite. Mon plus grand regret est de ne pas avoir mener le développement de l'application en TDD. En effet le débogage du front-end est complexe sans ces tests, et l'application en comporte un nombre significatifs :

- système de messages;
- gestion des erreurs;
- gestion de l'authentification;
- etc.

Pour la suite, je souhaite améliorer mes connaissances de ReactJS et revoir complètement le découpage du front-end. L'objectif et d'avoir une interface agréable est fonctionnelle pour les utilisateurs.

b. Back-end



Figure 3: SQLAlchemy

La mise en place du back-end a été plus complexe que je ne l'avais imaginé au début du projet. Le fait d'avoir utilisé SQLAlchemy lors de mon stage en entreprise, m'a aidé pour son utilisation. En revanche, la mise en place m'a demandé d'acquérir de nouvelles connaissances à mettre en pratique immédiatement.



Figure 4: FastAPI

FastAPI, à courbe d'apprentissage rapide. Mon expérience de l'utilisation du typage a été un vrai plus pour la compréhension de Pydantic et donc du système de données avec FastAPI. Le principal problème que j'ai rencontré : la gestion des utilisateurs. Django nous a habitué à une gestion native des comptes utilisateurs. Dans le cas de FastAPI, le développeur doit créer de toute pièce ce système. L'occasion pour moi de découvrir les bases de la création et de l'utilisation de token d'authentification. Autre point que je souhaite résoudre, l'utilisation d'un certificat SSL. Je maîtrise la configuration du serveur et la création du certificat. En revanche impossible d'avoir le

back-end fonctionnel lorsque le serveur est paramétré pour utiliser ce certificat. De manière générale, ce framework est une véritable bonne découverte. J'ai hâte de continuer à utiliser cet outil.

III.2. Planification du projet

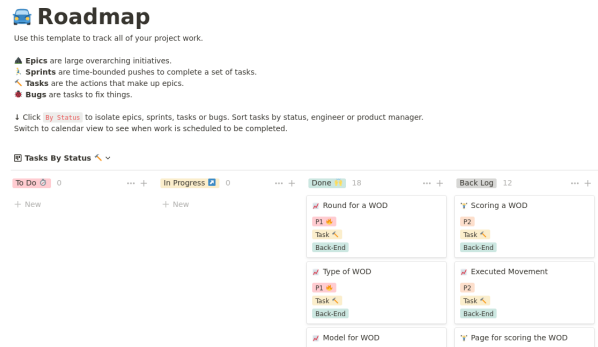


Figure 5: Roadmap

Pour la planification de ce projet, je me suis aidé de Notion. Par défaut, l'application propose un template de roadmap bien taillé. Il permet de créer des cartes sous forme de Kanban.

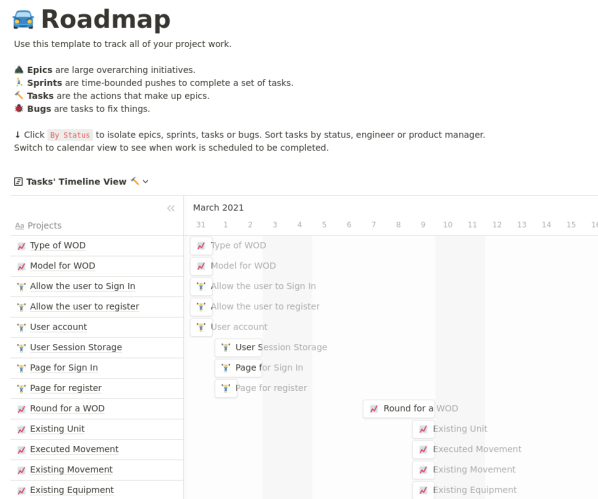


Figure 6: Time Line

Ces cartes peuvent être complétées par une date de début et de fin estimé. De cette façon on peut avoir une vue globale des tâches et de la chronologie du projet.

Les difficultés que j'ai rencontré sont les suivantes :

- oubli de mise à jour du statuts des tâches;
- aucune mise à jour de la chronologie lors de dérapage;

Cette organisation demande une discipline de fer pour en tirer le meilleur partie. Point sur lequel je dois encore progresser.

III.3. Fonctionnalités

Sur un projet comme celui-ci, le sujet nous tient à cœur. La conséquence directe est que l'on imagine un nombre important de fonctionnalités pour avoir l'application parfaite.

Puis la réalité nous rattrape assez vite. Un projet peut être analysé par un mélange de trois contraintes :

- temps alloué pour sa réalisation;

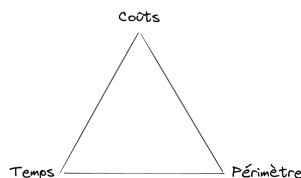


Figure 7: Triangle des contraintes

- coût de développement;
- périmètre du projet.

Dans le cadre de notre parcours, et dans la mesure où je suis mon propre financeur, le coût de développement peut être comparé à celui du coût mensuel de la formation. Plus je passais de temps sur le projet, plus celui-ci coûtait cher.

Au début du projet, j'ai défini un certain nombre de fonctionnalités qui me paraissaient être le minimum nécessaire. Soit le périmètre du projet.

J'ai aussi établie les dates, estimatives, de réalisation de chacune de ces fonctionnalités. Soit la contrainte de temps.

Ayant sous-estimé la difficulté de mise en place de la stack et le temps de développement de certaines fonctionnalités, le planning a rapidement dérapé. Le temps de m'en rendre compte, le coût du projet a augmenté puisque le temps de formation s'était allongé.

Pour résoudre ce problème et maîtriser la date de fin du projet, j'ai décidé de réduire le périmètre en plaçant une grande partie des fonctionnalités dans le backlog. L'application en est donc réduite à la création d'un compte utilisateur et la création d'un WOD.

L'objectif est de traiter les autres fonctionnalités dans le futur à l'issue de la formation.

Je retiens de cette expérience qu'il est primordiale de définir un cap. Mais aussi de régulièrement faire un point sur l'avancée et éventuellement corriger ce cap.

III.4 Bilan de projet

Il s'agit du premier projet conséquent que je mène de la conception à la réalisation. Le chemin pour le rendre fonctionnel et stable, est encore long. Mais le MVP est là !

Une nouvelle fois j'ai appris beaucoup de choses sur la mise en place d'une stack (back-end et front-end). J'ai pris de l'expérience sur la gestion et l'organisation d'un projet.

J'ai une grande satisfaction puisqu'il s'agit aussi de la conclusion de mon parcours de reconversion.