

# Dossier de conception technique

**EDEN CONCEPTION**

---

# Table des matières

## 1. Introduction

1. Objet du document
2. Références
3. Contexte

## 2. Architecture technique

1. Application web

## 3. Architecture de déploiement

1. Serveur de base de données
2. Serveur

## 4. Architecture logicielle

1. Principes généraux
2. Structure des sources
3. Diagramme de classe

# 1. Introduction

## 1.1 OBJECT DU DOCUMENT

---

Le présent document constitue le dossier de conception technique de l'application Eden Conception, représentant un système de gestion de projets.

## 1.2 RÉFÉRENCES

---

Pour obtenir de plus amples informations, merci de vous référer aux documents suivants :

1. DCF - Dossier de conception fonctionnelle de l'application

# 1. Introduction

## 1.3 CONTEXTE

---

L'application présentée est un projet pour la formation Développeur Python. Il doit s'agir d'un projet libre dont l'étudiant peut en définir les contours et les objectifs dans une note d'intention. Les technologies choisies sont libres. Le but de ce projet est qu'il doit avoir une portée sociale en s'adressant à une communauté d'utilisateur.

- Le projet doit donc répondre à un besoin d'un ensemble de bénéficiaires.
- Le projet doit apporter une amélioration, une bien-être à ses bénéficiaires.

J'ai donc choisi pour ce projet de réaliser une application web embarquant un système de gestion de projets. Ce projet s'applique à un public assez spécifiques (collègues développeurs, développeurs en entreprise ou même en free-lance) dont je vous parlerai juste après. L'application consiste donc à créer un compte afin d'avoir un tableau de bord. Celui-ci permet de créer un projet afin d'y incorporer des tâches. Ces tâches sont organisées dans des listes.

L'intérêt est qu'il est possible d'ajouter des collaborateurs à son projet. Ainsi, pour chaque tâches, le collaborateur peut inscrire le temps passé sur celle-ci et décrire ce qu'il a réalisé. Des statistiques sur le tableau de bord permettent au créateur du projet de suivre l'avancement de celui-ci dans un esprit d'organisation.

Une [version 2](#) de l'application permettra aux utilisateurs de communiquer par un système de messagerie dans chaque détails de tâches (en cas de changement de besoin ou autre).

# 2. Architecture technique

## 2.1 APPLICATION WEB

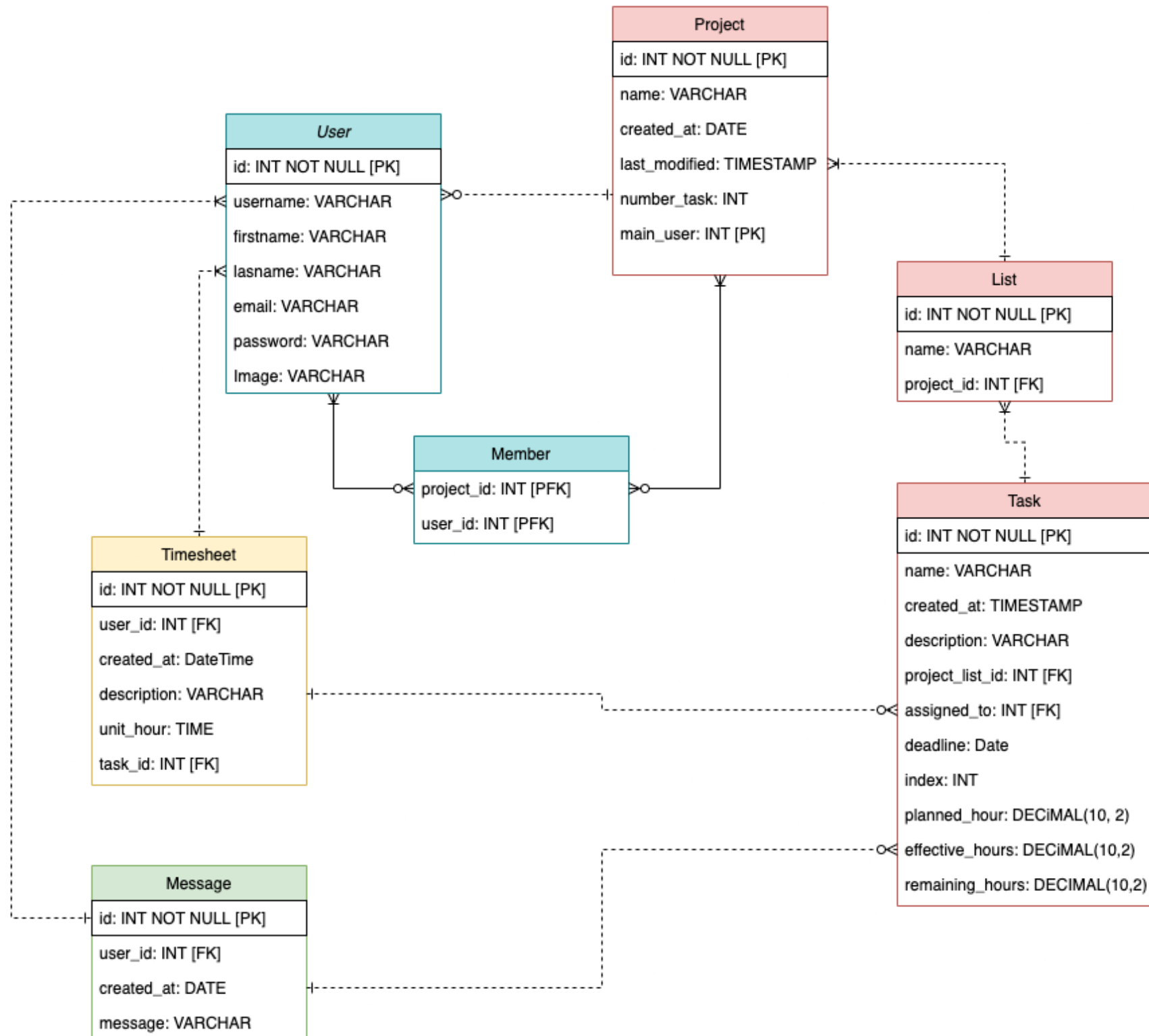
---

L'application sera créée avec le framework [Django](#) (3.2) développé avec le langage [Python](#) (3.8). Nous utiliserons la librairie [jQuery](#) qui nous permettra de concevoir des fonctionnalités de manière asynchrone. Des statistiques seront intégrées à chaque projets créés. Nous utiliserons donc la librairie JavaScript : [AmChart](#) pour l'affichage. Enfin, nous incorporons le pré-processeur [Sass](#) afin de pouvoir écrire du css en scss.

# 3. Architecture de déploiement

## 3.1 SERVEUR DE BASE DE DONNÉES

Voici le modèle physique de données du futur système. La base de données utilisée sera [PostgreSQL](#).



# 3. Architecture de déploiement

## 3.2 SERVEUR

---

L'application sera hébergée sur un serveur du service d'hébergement [DigitalOcean](#). Le serveur sera configuré avec un 1er serveur web HTTP : [Nginx](#), puis un 2ème : [Gunicorn](#) qui n'est autre qu'un serveur HTTP Python utilisant les spécifications WSGI (*Web Server Gateway Interface*). Celui-ci nous permettra de communiquer avec l'application Django. Il sera contrôlé par [Supervisor](#), un système de contrôle des services d'un environnement UNIX.



# NGINX

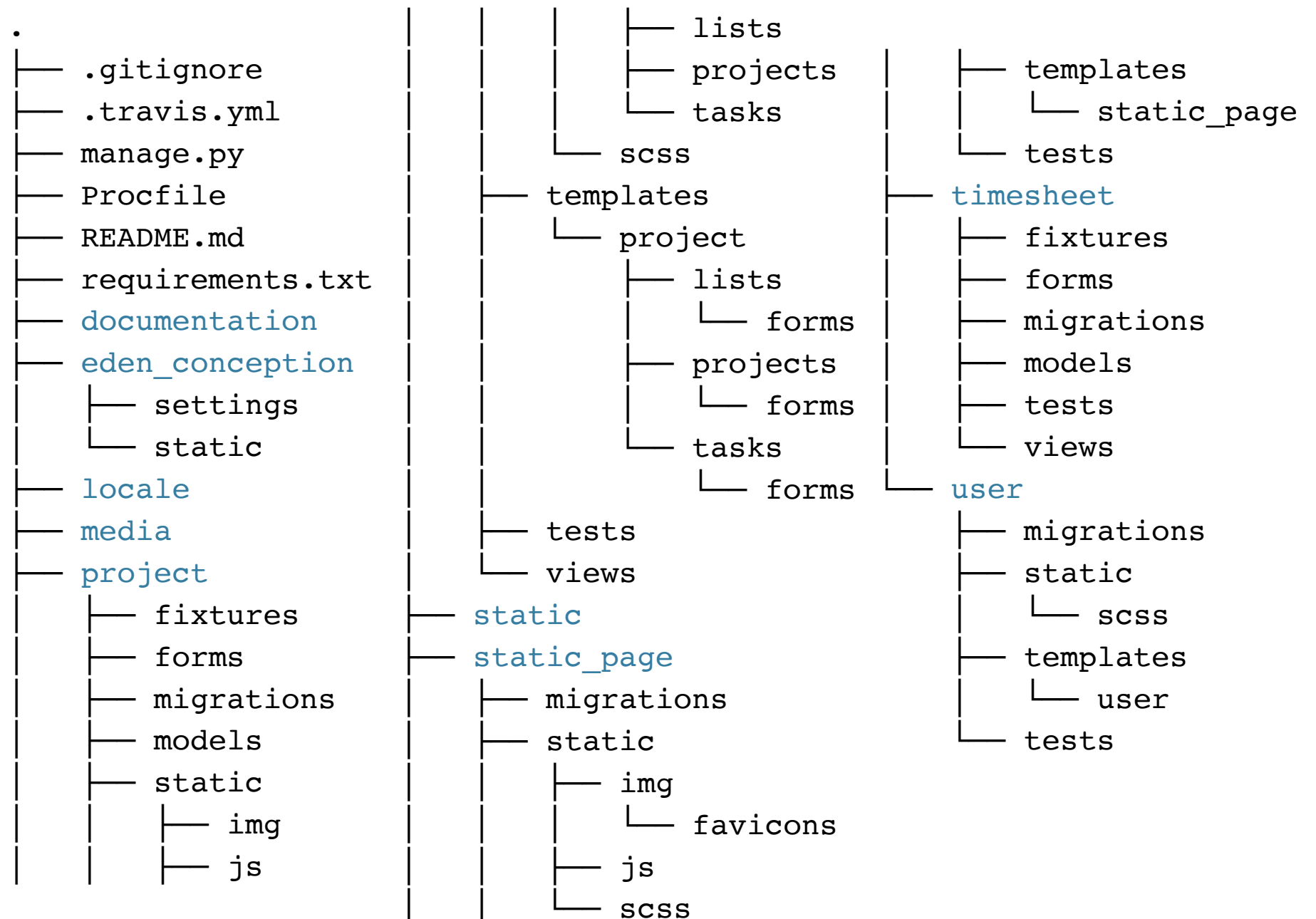
# 4. Architecture logicielle

## 4.1 PRINCIPES GÉNÉRAUX

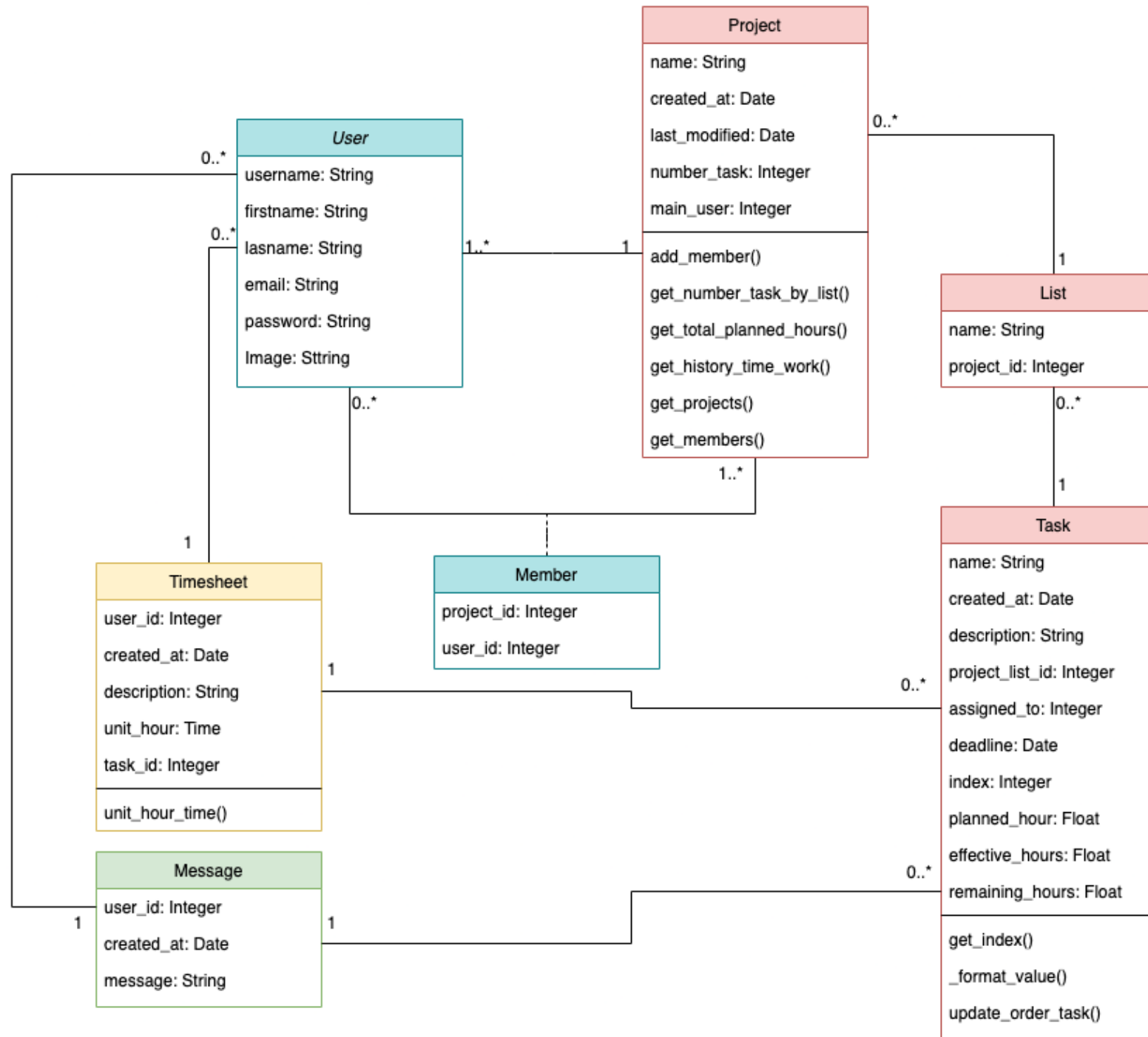
Les sources et versions du projet seront gérées par Git et GitHub.

## 4.2 STRUCTURE DES SOURCES

La structuration des répertoires du projet suit la logique de l'architecture **MVT** du framework Django.







Fin