GERER-MON-BUDGET

Dossier de conception technique

Version 1.1

Auteur
FERRANDEAU MATHIEU
Développeur

Table des matières

1 - Versions

Auteur	Date	Description	Version
MF	07/02/2020	Création du document	1.0
MF	07/02/2020	Mise a jour du document	1.1

2 - Introduction

2.1 - Objet du document

Le présent document constitue le dossier de conception technique de l'application **GERER-MON-BUDGET**.

L'objectif du document est de présenter les outils, les technologies et les méthodes mises en oeuvre pour réaliser l'application.

Les éléments du présent dossier découlent :

- de l'analyse des besoins et de la rédaction du dossier de conception fonctionnelle.

2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants:

1. **DCF – 1.1** : Dossier de conception fonctionnelle de l'application
2. **DE – 1.1** : Dossier d'exploitation de l'application.

3 - Architecture Technique

3.1 - Composant Généraux

3.1.1 - Package Registration:

3.1.1.1 - Composant S'enregistrer :

Ce composant permet aux utilisateurs de se créer un compte sur le site.

3.1.1.2 - Composant Se connecter:

Ce composant permet aux utilisateurs de s'authentifier sur le site.

3.1.1.3 - Composant Mon profil:

Ce composant permet aux utilisateurs de modifier leurs informations.

3.1.1.4 - Composant Deconnexion:

Ce composant permet aux utilisateurs de se déconnecter.

3.1.2 - Package Spent:

3.1.2.1 - Composant Enregistrer dépense :

Ce composant permet à un utilisateur authentifié d'enregistrer des dépenses.

3.1.2.2 - Composant Historique :

Ce composant permet à un utilisateur d'accéder à toutes les dépenses qu'il a enregistré au préalable et lui permet de les modifier/supprimer s'il le souhaite.

3.2 - Application web

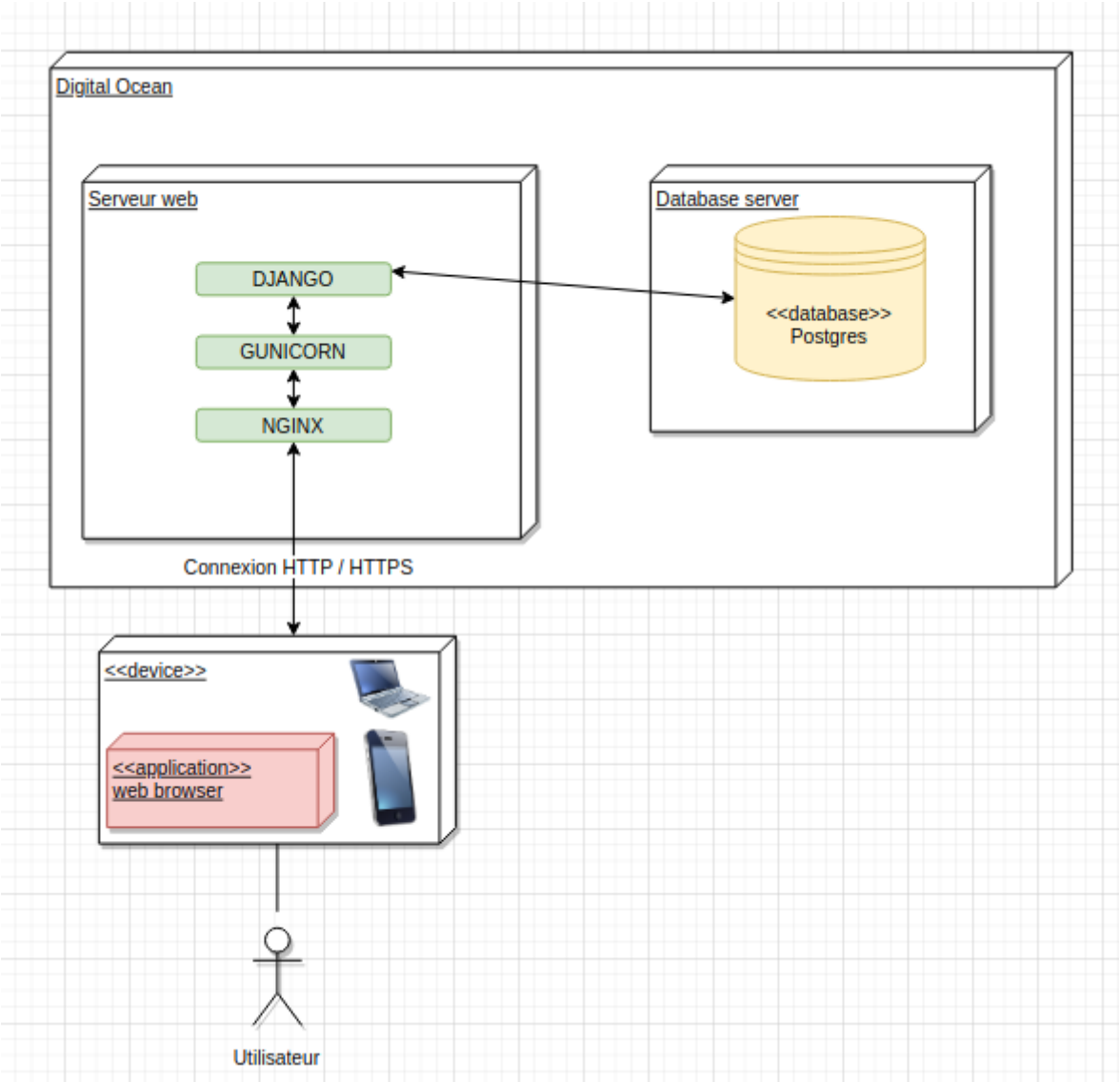
Le serveur est hébergé par la plateforme Digital Ocean.

La pile logicielle est la suivante:

- Serveur d'Application **Django** (V3.0.2) / **Python** (V3.6.9) / **Gunicorn** (V19.9)
- Serveur Web **Nginx** (V1.14)
- Serveur de base de données **Postgres** (V12.1)

4 - Architecture de Déploiement

4.1 - Déploiement de l'application



4.2 - Serveur de Base de données

Le serveur de base de données permet de stocker les données de l'application web.

Caractéristiques techniques du serveur:

OS : Ubuntu.18.04.3

SGBD : PostgreSql

RAM : 1GO

STOCKAGE : 25GO

4.3 - Serveur Web

Le serveur Web permet l'interaction entre l'utilisateur et la base de données et permet aussi l'affichage du site web.

Caractéristiques techniques du serveur:

OS : Ubuntu.18.04.3

Serveur HTTP: Nginx

RAM : 1GO

STOCKAGE : 25GO

5 - Architecture logicielle

5.1 - Principes généraux

5.1.1 - Applications Django

Le développement de l'application **GERER-MON-BUDGET** se présente sous la forme d'un projet Django, lui même divisé en trois « applications » : Core, Registration et Spent.

Chaque application respecte le pattern Model Vue Template (MVT).

5.1.2 - Les couches

L'architecture de chacune des trois applications est la suivante :

- une couche **Template**: responsable de l'interface utilisateur/application.

- une couche **Vue**: responsable de la logique métier et de la manipulation des données.

- une couche **Model** : responsable de la représentation des données.

Exemple de découpage en couches de l'application «Spent» :

Couche	**Rôle**
Template	Regroupe les notions de Views et Templates de l'application. Elle représente la partie servant d'interface entre l'utilisateur et l'application.
Vue	Regroupe les fonctions et algorithmes nécessaires au fonctionnement de l'application tels que : - L'affichage de l'historique des dépenses enregistrées au préalable. - Choix d'une dépense en fonction de sa catégorie.
Model	Regroupe les classes utilisées par l'application «Spent» pour manipuler les données (ex : Category, Outlay, UserOutlay)

5.1.3 - Les modules

Modules nécessaires à l'application

- asgiref

- atomicwrites

- certifi

- chardet

- coverage

- Django

- gunicorn

- idna

- isort

- mccabe

- more-itertools

- psycopg2

- py

- pyparsing

- pytz

- requests

- six

- sqlparse

- typed-ast

- urllib3

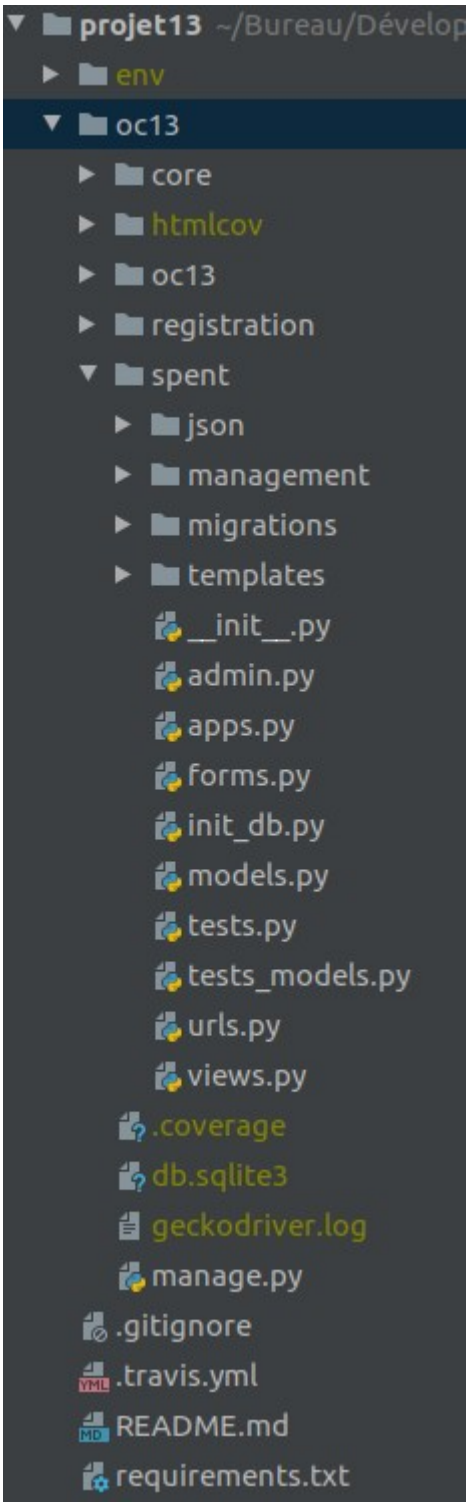
- wcwidth

- wrapt

- zipp

5.1.4 - Structure des sources

La structuration des répertoires du projet est faite de façon à respecter les bonnes pratiques d'un projet Django et elle suit donc la logique suivante :



6 - Points particuliers

6.1 - Environnement de développement

Le développement de l'application ne requiert pas l'utilisation d'un IDE. Elle sera développée en utilisant un éditeur de texte au choix des développeurs (Pycharm).

7 - Glossaire

DJANGO	Framework permettant la réalisation web en Python
IDE	Environnement de développement intégrée
SGBD	Système de gestion de base de données