# Petits comptes entre amis

Steven Liatti

Développement et services web - Prof. Stéphane Malandain

Hepia ITI 3ème année

17 octobre 2017

# Table des matières



# Table des figures



# Table des listings de code source

# 1 Introduction

## 1.1 Description

Le but de ce mini-projet est de réaliser un site en php permettant à des amis de noter et partager les dépenses effectuées par et pour le groupe au cours de vacances communes. Lorsque lune des personnes fait des courses, par exemple, elle lenregistre. Chacun enregistre les dépenses qui concernent le groupe. Ainsi, à la fin du séjour (ou à tout moment) on peut savoir qui a payé quoi et surtout ce que chacun doit aux autres personnes du groupe damis.

### 1.2 Technologies utilisées

- Base de données :
  - MySQL, avec
  - MySQL Workbench (pour la création du schéma)
- Back-end :
  - Apache, serveur HTTP
  - PHP, avec
  - Silex, micro framework PHP basé entre autres sur Symfony
- Front-end :
  - Twig, moteur de templates pour PHP (utilisé de concert avec Silex)
  - jQuery
  - Bootstrap pour le design en CSS

## 2 Base de données

Les technologies imposées pour la base de données de ce travail pratique sont SQLite ou MySQL. J'ai choisi d'utiliser MySQL, car je suis familier avec. J'ai profité de cette occasion pour découvrir et utiliser Workbench, un programme permettant de modéliser les tables et relations d'une base de données de manière graphique. Une fois le model terminé, Workbench offre la possibilité de l'exporter en instructions SQL (création de tables et contraintes).

Mon schéma est constitué des tables suivantes :

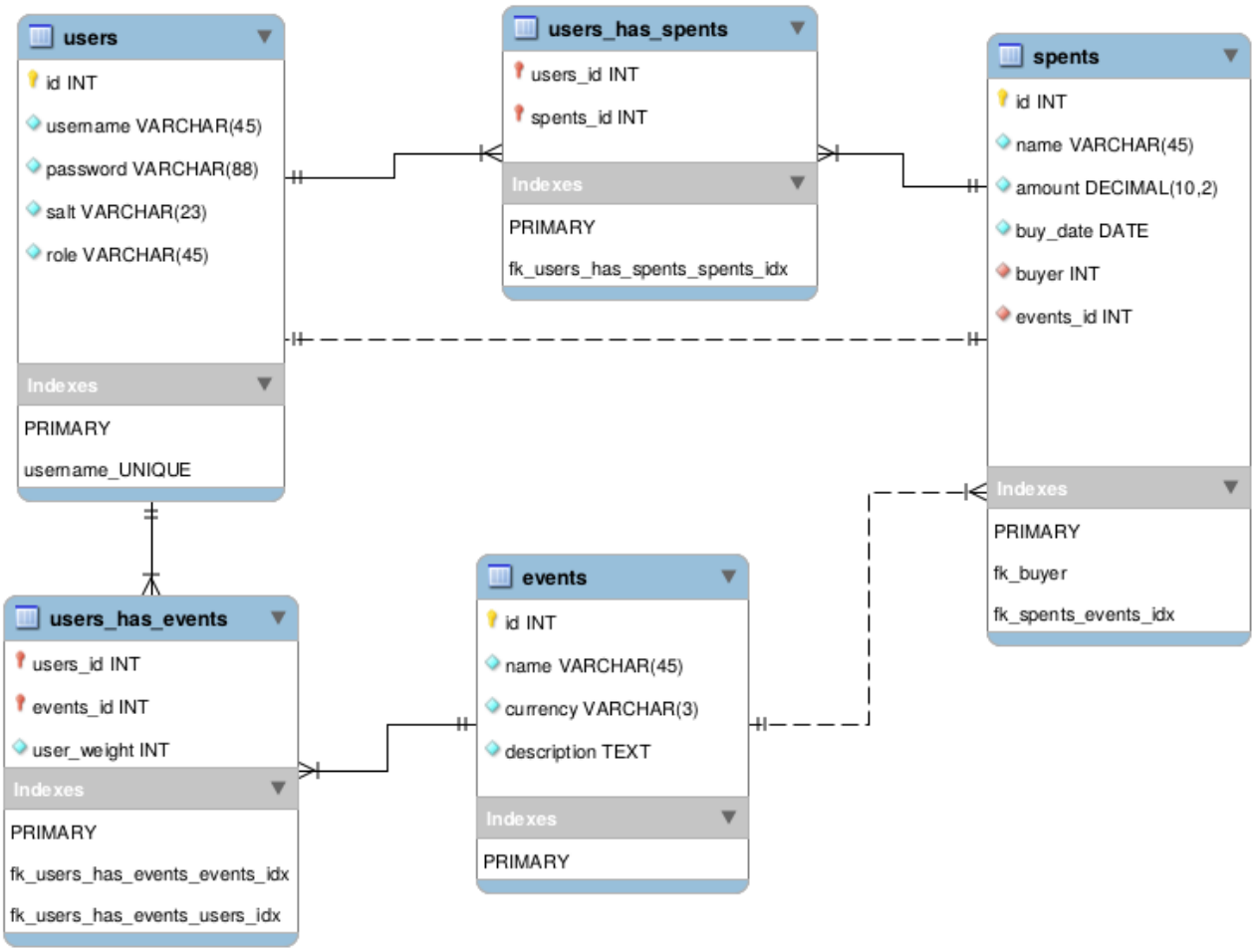


FIGURE 1 – Schéma de la base de données relationnelle

Il y a 3 tables principales : les utilisateurs, les événements et les dépenses. 2 autres tables secondaires font la liaison entre les utilisateurs et les événements et les utilisateurs et les dépenses respectivement (liaison Many-To-Many). La table des utilisateurs possède un champ `salt` et un autre `role`, ils sont nécessaires au fonctionnement de Silex (explications plus loin). Le poids de chaque utilisateur au sein d'un événement est indiqué dans la table croisée `users_has_events`. Chaque dépense référence l'acheteur (dans `users`) et l'événement lié (dans `events`). Ce schéma représente l'interface minimum pour les données de ce travail, mais il a l'avantage d'être simple à comprendre et à maintenir.

# 3 Back-end

Je profite également de ce TP pour appréhender Silex, un micro framework PHP dérivé de Symfony (que j'ai eu l'occasion de tester), beaucoup plus léger que son grand frère mais tout de même robuste et modulaire. Il bénéficie d'un grand nombre de modules à ajouter, en vrac : système de templates, connexion à la base de données, routes, etc.

## 3.1 Architecture MVC

Grâce à Silex, mon architecture respecte le design pattern MVC. Voici l'arborescence du site :

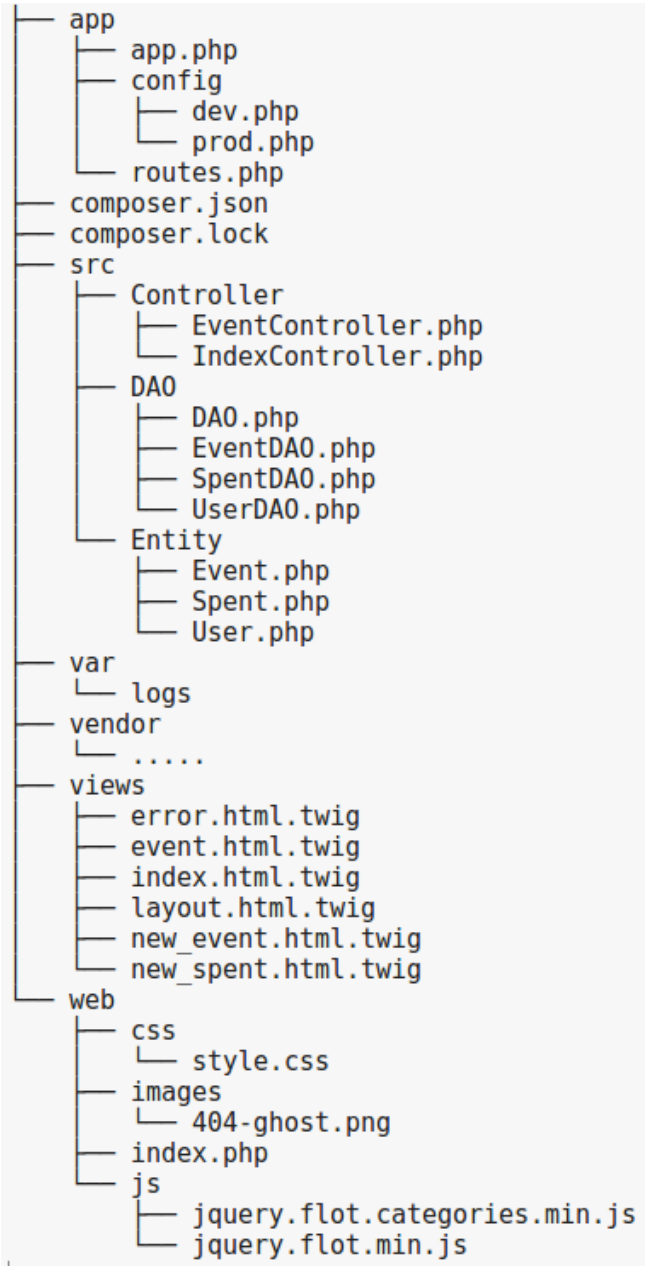


FIGURE 2 – Arborescence du site

# 4 Front-end

# 5 Conclusion

## 5.1 État actuel du projet

## 5.2 Propositions d'améliorations

— Tests unitaires

— Dans une situation réelle de production, on ne laisserait pas un utilisateur choisir parmi tous les autres utilisateurs au moment de créer un nouvel événement. Il faudrait plutôt ajouter des adresses email, d'utilisateurs déjà inscrits (ou non) sur le site, ainsi chaque invité peut rejoindre l'événement.

— Changer le code lié à la sélection du poids de l'utilisateur (lors d'un nouvel évenement) pour l'intégrer avec le composant Form de Symfony.