



Together Organising and Managing Activities for Travelling Overseas

Réalisé par

Assil El Yahyaoui

Sous la direction de

Anne LAURENT | Arnaud CASTELLTORT

Année universitaire 2016-2017

Sommaire

1 Cahier des charges	3
1.1 Présentation du Projet	3
1.2 Analyse des besoins fonctionnels	3
1.3 Contraintes ergonomiques	5
2 Conception	5
2.1 Base de données	5
2.1.1 Choix de la base de données	5
2.1.2 Modèle Conceptuel de données	6
2.2 L'architecture MVC	6
2.3 Langages utilisés	7
2.3.1 Langage côté Client :	7
2.4 Architecture de déploiement	8
3 Rapport d'activité	9
3.1 Difficultés	9
3.2 Avantages	9
3.3 Ce que j'aurais fait différemment	9
4 Conclusion	10

1 Cahier des charges

1.1 Présentation du Projet

L'application web que j'ai réalisée permet d'organiser de façon démocratique un séjour ou un voyage au sein d'une famille ou un groupe d'amis. En effet, généralement les voyageurs ont des loisirs différents et aimeraient donc faire des activités différentes.

Le projet permettrait à chaque membre de proposer une idée, activité ou destination, qui sera mise au vote par les autres participants. Une note globale sera donc attribuée pour chaque proposition. Ceci permettra d'avoir les grandes lignes du voyage qui puisse satisfaire les besoins des personnes concernées.

Les dépenses peuvent également être suivies grâce à un onglet qui fournit le budget estimé pour le voyage.

L'application est nommée T.O.M.A.T.O qui est un acronyme pour la phrase suivante : " Together Organising and Managing Activities for Travelling Overseas". L'application a été créée en anglais pour atteindre le plus de personnes possible notamment au sein de mon entourage.

1.2 Analyse des besoins fonctionnels

Pour pouvoir accéder au site, l'utilisateur devra d'abord s'y inscrire. Il doit par la suite créer un groupe de personnes ou joindre un groupe qui a été fait par une autre personne. Pour assurer la sécurité et la confidentialité du groupe, un mot de passe est attribué. Une fois que ces deux paramètres sont choisis, des activités peuvent être insérées.

Il est possible de changer de groupe ou de voyage à tout moment grâce aux boutons de la barre de navigation. Toute activité ajoutée n'est accessible que par les membres du groupe où elle a été créée.

Chaque activité appartient à une catégorie (transport, nourriture etc...). On peut ainsi visualiser les dépenses par catégorie et les adapter en fonction de nos priorités.

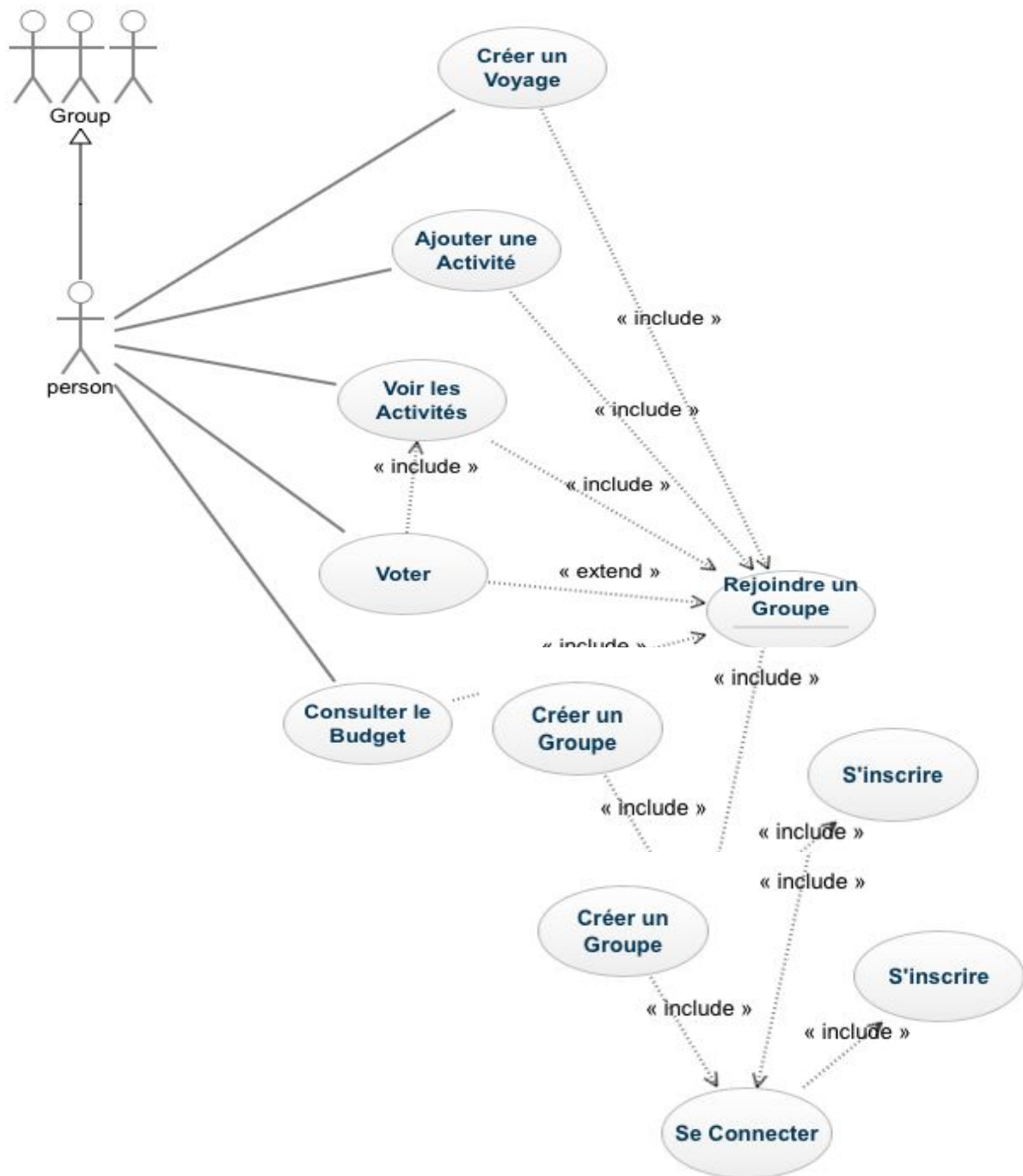


Illustration 1 : Diagramme de cas d'utilisation

1.3 Contraintes ergonomiques

Vu que cette application web est destinée à simplifier la gestion d'un voyage, il est impératif qu'elle soit accessible non seulement sur ordinateur portable mais également sur téléphone et tablette. De plus, son utilisation doit être facile d'utilisation car elle est destinée aux personnes de tout âge. Le but est bien évidemment de prendre l'avis de tous.

2 Conception

2.1 Base de données

2.1.1 Choix de la base de données

Après comparaison via des sites internet, le choix de la base de données s'est porté sur une base de données relationnelle (SQL) plutôt que sur un système de gestion de type NoSQL. En effet, ses avantages sont multiples : elle est facile d'accès et sa création est relativement simple. De plus, il est possible de mettre en oeuvre des requêtes complexes comme les croisements multiples de données .

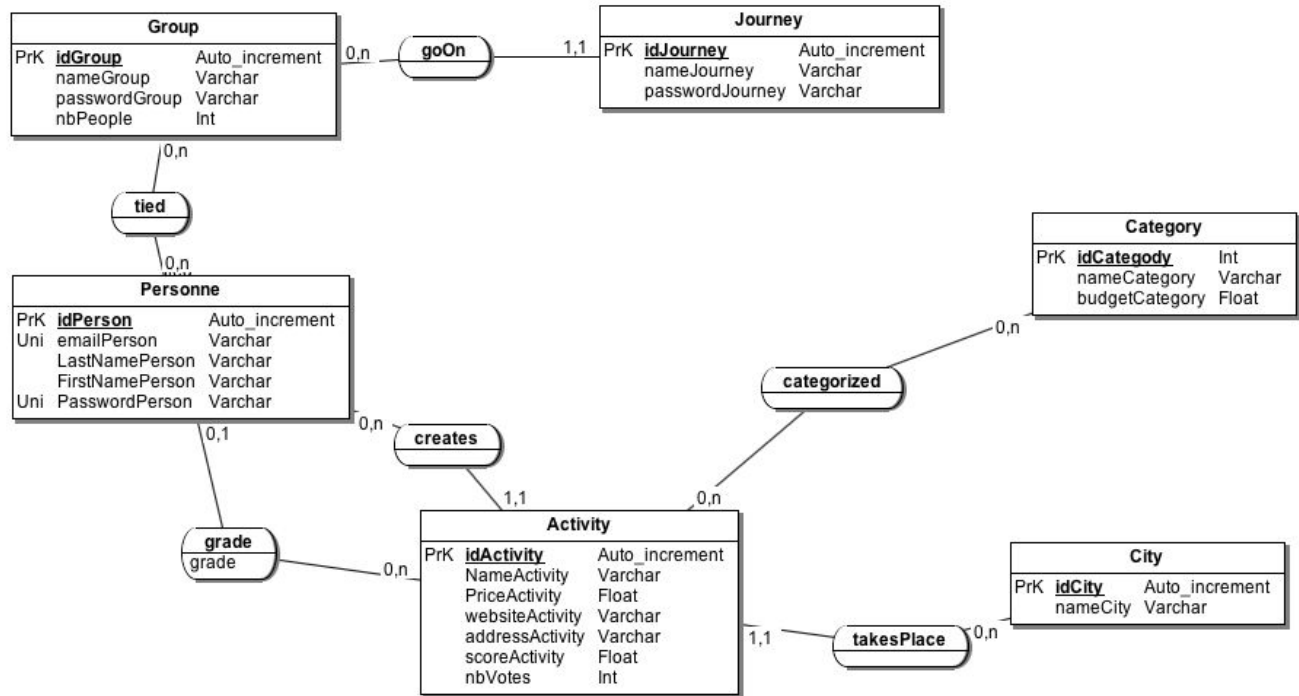
La base de données relationnelle a néanmoins des inconvénients: sur un très grand volume de données (centaines-milliers de Teraoctets) le modèle peut atteindre des limites en matière de performance. Ceci n'est pas le cas en NoSQL qui est conçu pour de très grands ensembles de données distribuées (par exemple quand une entreprise doit accéder à des fins d'analyse ou à de grandes quantités de données).

Le système de gestion de bases de données utilisé lors de la conception du projet est PostgreSQL. Ce choix a été fait non seulement car je m'y suis familiarisé lors des cours suivis cette année mais également pour le fait qu'il présente beaucoup de qualités: il est orienté objet contrairement à MySQL, il présente une interface graphique plus évoluée que cette dernière.

PostgreSQL permet également de gérer des données en grande quantité et il peut mieux contrôler la base de données. Son seul désavantage réside dans le fait que sa documentation sur internet est moins populaire que celle de MySQL

2.1.2 Modèle Conceptuel de données

Ci-dessous le modèle conceptuel de données de mon application



MCD du projet TOMATO

Le script de création peut être consulté ici :

<https://github.com/assilelyahyaoui/ Tomato/blob/master/script%20creation%20table>

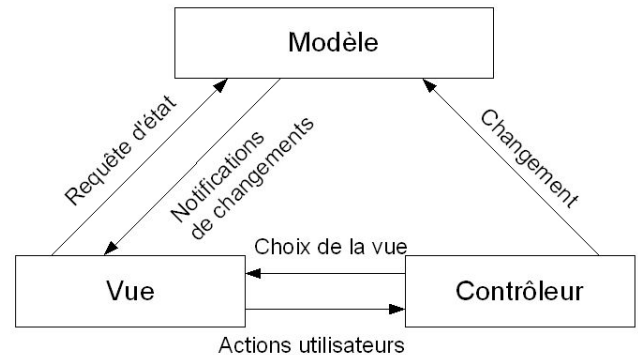
2.2 L'architecture MVC

Le pattern utilisée lors du développement du site est le Model-View-Controller. Ces trois modules ont des fonctionnalités différentes:

- Modèle : est responsable de récupérer les données.
- Vue: contient la présentation de l'interface graphique et retourne une présentation des données provenant du Modèle.

- Un contrôleur: contient la logique concernant les actions effectuées par l'utilisateur et est responsable de retourner une réponse avec l'aide mutuelle des couches Model et Vue.

Le schéma ci-contre représente parfaitement les interactions entre les 3 modules :



Ce pattern permet donc de séparer la représentation des données côté client de la partie métier (Modèle). La vue et le contrôleur s'occupent quant à eux de la première partie.

2.3 Langages utilisés

2.3.1 Langage côté Client

Les langages côté Client améliorent l'expérience côté utilisateur en rendant le site dynamique. Voici ceux que j'ai utilisés :

- HTML : c'est un langage qui permet de retourner des ressources en donnant du sens. Ce langage fait de balise permet donc de présenter l'information. Les balises permettent de structurer la page et mettre en forme le texte et de placer des éléments interactifs, tels des liens, des images ou bien encore des animations. C'est le langage obligatoire pour créer des sites web et donc ne peut être substitué.
- CSS: permet la mise en forme du site. Il est pratique puisqu'il permet de changer des balises similaires dans le fichier HTML ou XML en modifiant une seule partie du CSS.

- Javascript : j'ai également utilisé du javascript avec la bibliothèque jQuery. Cette dernière permet de faciliter l'écriture de scripts côté client.

2.3.2 Langage côté Serveur:

Le langage utilisé côté Serveur est : PHP

C'est un langage qui est très facile à apprendre et à pratiquer. De plus la documentation et la communauté sont omniprésentes sur internet; ce qui est très utile lorsqu'on essaye de déboguer quand il s'agit d'apprendre à coder.

PHP est orienté vers la programmation pour les sites web mais permet de faire bien d'autres usages; ce qui pourrait être pratique pour le futur.

Du fait que PHP soit vieux, il est moins performant que d'autres langages comme Node.JS

2.4 Architecture de déploiement

Pour pouvoir déployer mon site, j'ai choisi d'utiliser l'hébergeur HEROKU.

C'est un PaaS (Platform As A Service) : il propose une plateforme de développement d'application web en fournissant l'infrastructure matérielle ainsi que les briques logicielles nécessaires au démarrage de tout projet.

Il propose également une solution d'hébergement gratuite. De plus; il peut être relié directement à GitHub (un service web d'hébergement et de gestion de développement de logiciels).

Relier GitHub a HEROKU permet le déploiement automatique lorsqu'on fait des 'push'.

3 Rapport d'activité

3.1 Difficultés

TOMATO est la première application web que je crée. De ce fait, j'ai dû apprendre plusieurs langages en très peu de temps. Le manque de maîtrise a causé des problèmes notamment lors de la connexion au PDO et du déploiement. Le temps attribué au débogage fut énorme à cause de mes lacunes.

Beaucoup de mes objectifs initiaux n'ont pas été atteints: mon idée de base était de tout incorporer sur un calendrier en ajoutant les activités par date. En cas de conflit de créneaux, l'activité avec le moins de votes aurait été automatiquement effacée une fois que toutes les personnes du groupe aient voté.

3.2 Avantages

Comme j'étais contrainte par le temps, il fallait que je sois le plus efficace possible. Ce projet m'a donc non seulement appris de nouveaux langages mais m'a également inculqué une nouvelle façon de réfléchir et d'apprendre par soi-même. C'est un projet que je reprendrai dans le futur proche pour l'améliorer et lui ajouter des fonctionnalités.

3.3 Perspectives d'évolution

Comme évoqué précédemment, j'ai attribué beaucoup de temps à l'apprentissage des différents langages tels que PHP ou encore AngularJS (que je n'ai finalement pas utilisé en raison de bugs que je n'ai pas pu résoudre).

J'aurais dû également être plus raisonnable dans mes objectifs en faisant moins mais mieux ainsi que répartir les tâches à l'avance et me mettre des contraintes de temps.

4 Conclusion

Ce projet m'a énormément enrichie au niveau de la pratique, de l'organisation et de la prise de décision.

Grâce à ce travail, j'ai pu acquérir de nouvelles connaissances qui me serviront dans le futur.

Mon projet est consultable au lien suivant :

<https://github.com/assilelyahyaoui/Tomato>