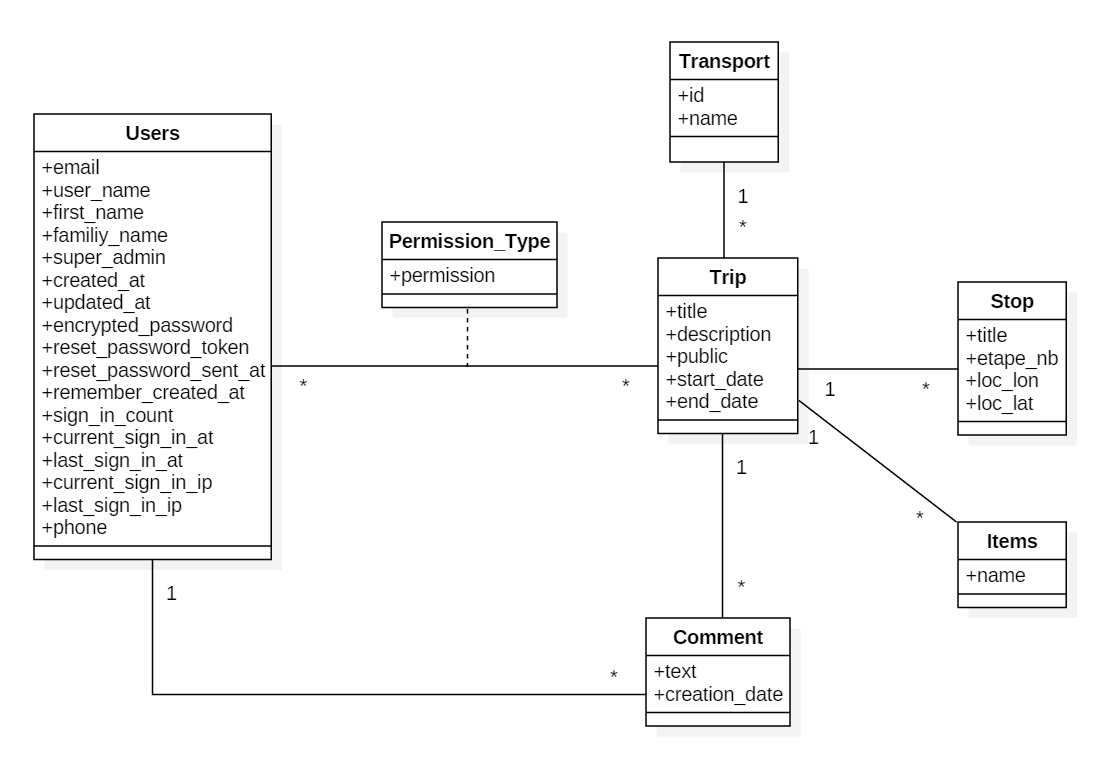
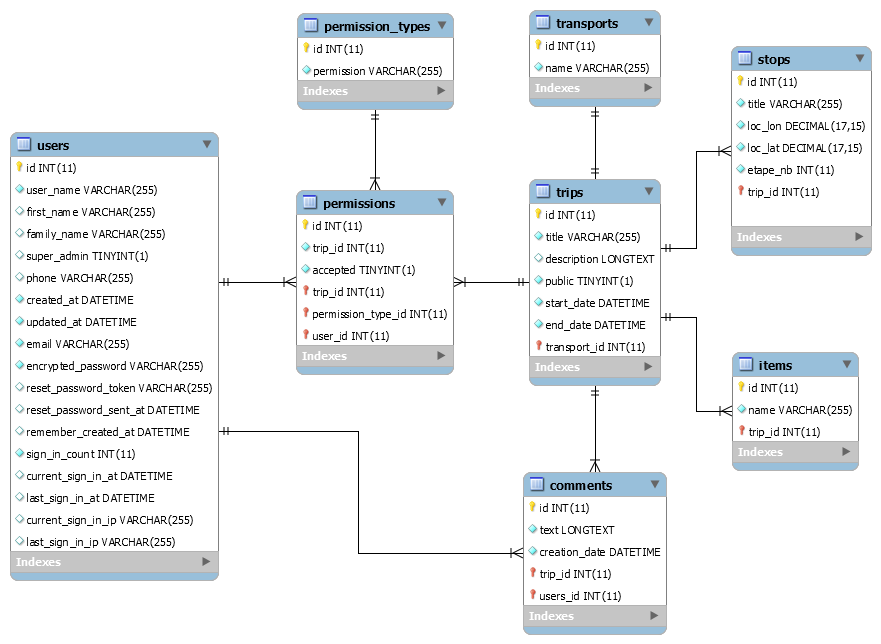
# Structure de la base de données

## Schéma conceptuel



## Schéma relationnel



## Description

Notre site possède 6 entités conceptuelles qui se représentent en 8 tables SQL. La table trip est au centre du projet, elle est connectée à toutes les autres tables à travers différentes relations. A noter que la table « User » est reliée à travers une table d’association « permission » qualifiant chaque appartenance d’un utilisateur à un voyage par un niveau de droit allant de simple participant à admin. Cette même table d’association est relié à une table contenant les différents types de permissions.

La table « Stop » contient les différentes étapes d’un voyage. Ces étapes ont un ordre et sont déterminées par des coordonnées. « Comments » contient les commentaires des participants d’un voyage. « Items » est constitués de la « packing-list » d’un voyage. « Transports » désigne les différents modes de transport disponibles lors de la création du voyage.

Contraintes d’intégrité :

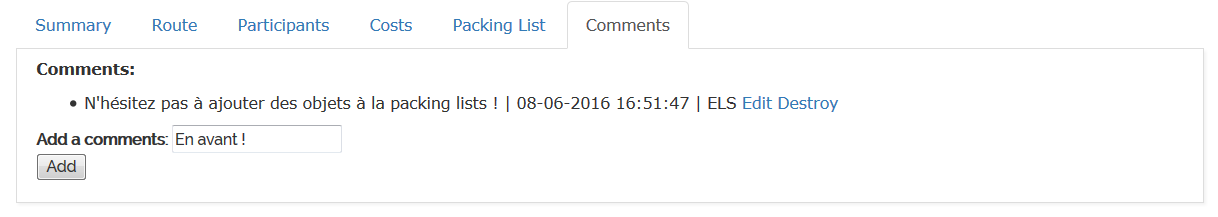
Les associations entre les tables se font via les champs « id » qui jouent le rôle de clé primaire et des « XXX\_id » qui jouent le rôle de clés étrangères. Les liaisons se font concrètement au sein de l’application Ruby on Rails dans la définition des modèles.

Les champs «NOT NULL » sont représentés par des losanges pleins alors les champs facultatifs par des losanges vides. Les id sont bien sûres uniques.

# Implémentation

## Description de la partie Ajax

Nous avons utilisé Ajax pour l’ajout de commentaires et d’objets dans la packing list :

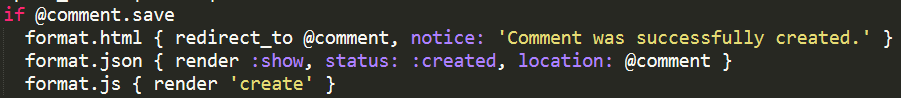


Lors de l’ajout d’un nouveau commentaire, ce dernier apparait automatiquement dans la liste et est stocké dans la DB sans rafraîchir la page dans sa totalité. La gestion des objets de la packing-list suit la même logique.

Afin que le formulaire d’ajout de commentaire ne charge pas une nouvelle page lors du submit, on indique la propriété remote => true



Ensuite, dans le controller gérant la création de commentaire, on y ajoute la ligne format.js :



Cela va exécuter le fichier create.js.erb déterminant l’action à effectuer en plus de l’ajout dans la DB :

$**(**'#comment\_list'**).**append**(**"<li> <%= @comment.text %> | <%= @comment.creation\_date.strftime('%d-%m-%Y %H:%M:%S')%> |" **+** $**(**'#user\_name'**).**html**()** **+** "<%= j(link\_to 'Edit', edit\_comment\_path(@comment)) %> <%= j(link\_to 'Destroy', @comment, :confirm => 'Are you sure?', :method => :delete, :remote => true, :class => 'comment') %></li>"**);**

Ce fichier permet l’ajout du nouveau commentaire avec l’affichage de l’heure, de l’utilisateur ainsi que des liens pour éditer et supprimer ce dernier

# Etats des lieux

La plupart des fonctionnalités ont été implémentées. Voici les fonctionnalités partiellement implémentées :

## Calcul du coût en carburant et en péages

Nous avions initialement prévu d’utiliser la REST API fournie par Michelin afin de calculer automatiquement les coûts en carburants et éventuels péages sur le trajet. En raison des contraintes pour l’obtention d’un compte et des délais d’attentes, nous avons préféré utiliser l’API fournie par Google Maps qui est plus limitées mais plus facile d’utilisation. Elle permet le calcul d’itinéraire pour des trajets en vélo et à pieds, ce que ne permet pas Michelin.

Nous permettons tout de même à l’utilisateur de fournir un coût par kilomètres pour ainsi calculer dynamiquement un coût pour le trajet sélectionnés.

## Demande d’adhésion à un trip publique

Il n’est pas actuellement possible pour un utilisateur de demander à joindre un trip public. Il reçoit bien une invitation lorsqu’il est ajouté à un trip mais cela ne va pas dans l’autre sens.

## Quitter un trip

La possibilité de quitter un trip n’a pas été implémentée par manque de temps. La seule manière de le faire pour l’instant est de supprimer totalement un trip.

# Conclusion

Ce projet nous a permis de mieux comprendre le fonctionnement de Ruby on Rails. Bien que pendant les laboratoires nous avons déjà pu intégrer la majorité de la matière, c’est en appliquant dans un cas pratique sans contrainte que nous avons pu intégrer réellement le concept.

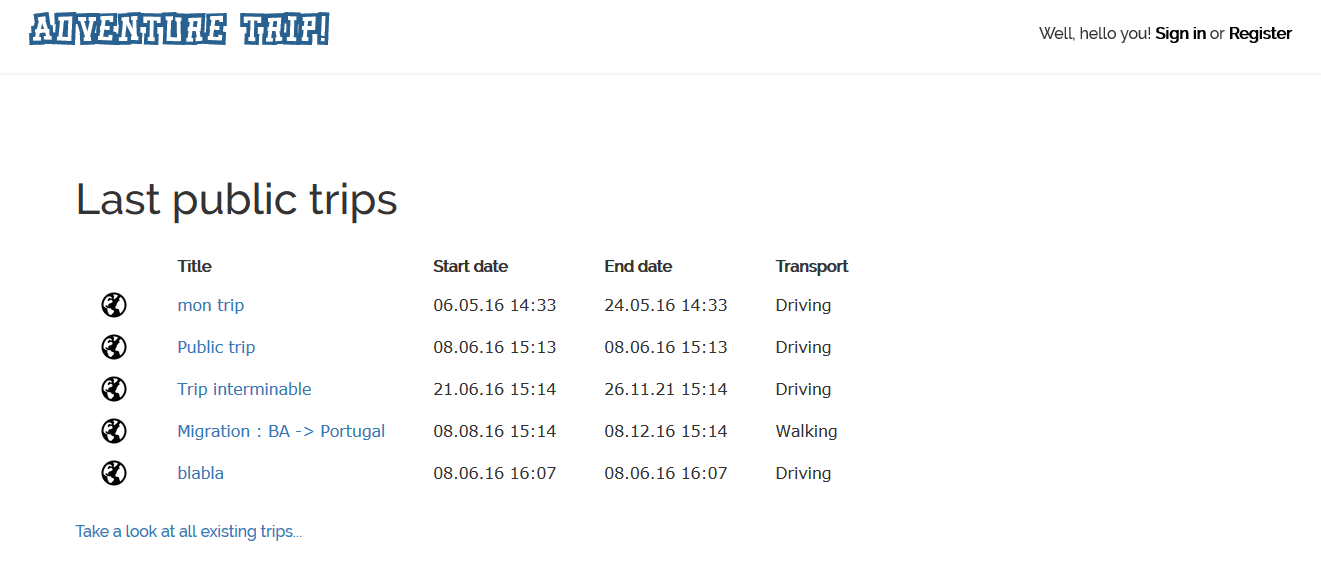
Nous estimons avoir fait un bon travail bien que pas parfait. Principalement par manque de temps, certaines fonctionnalités sont manquantes mais le site est complètement fonctionnel en l’état.

# Annexe

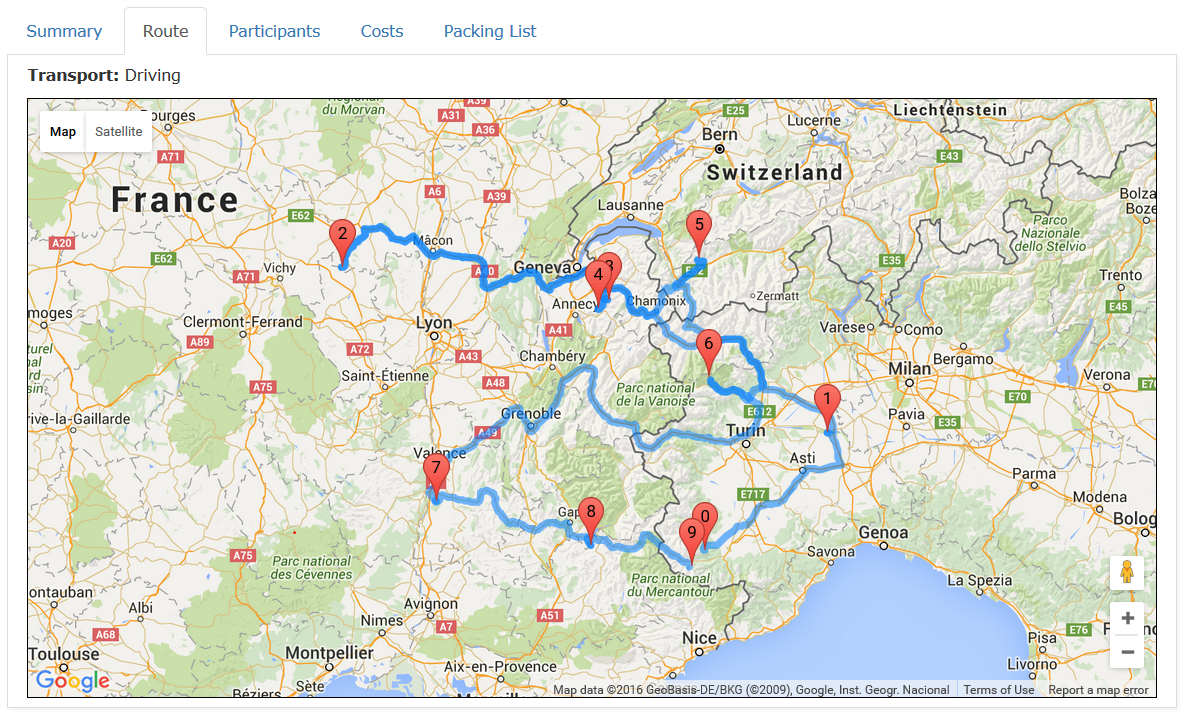
## Mode d’emploi

# Mode d’emploi

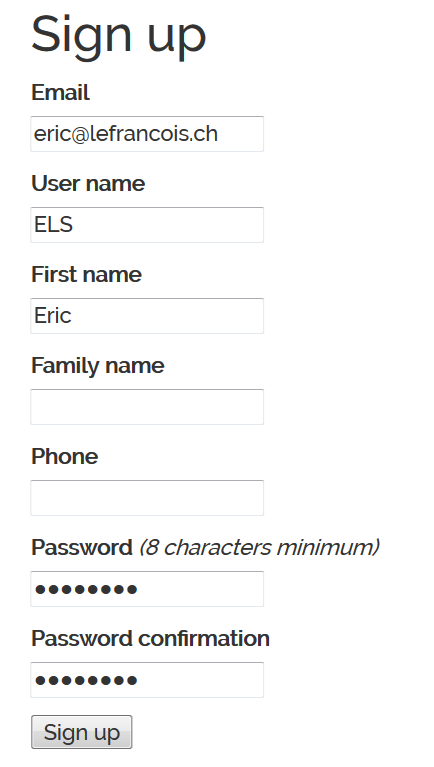
Une fois arrivé sur le site, un simple visiteur peut uniquement consulter les trips publics.



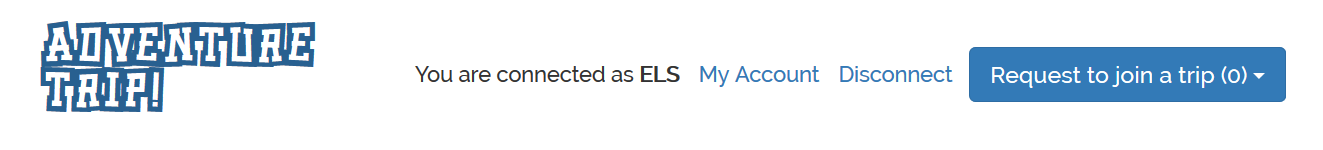
Il peut par exemple voir l’itinéraire de « mon trip » en cliquant sur ce dernier.



Mais si veut créer ou être ajouter dans un trip, il lui faut donc créer un compte via le lien « Register ». Il doit fournir des informations, au minimum son email, un nom d’utilisateur, ainsi qu’un mot de passe.

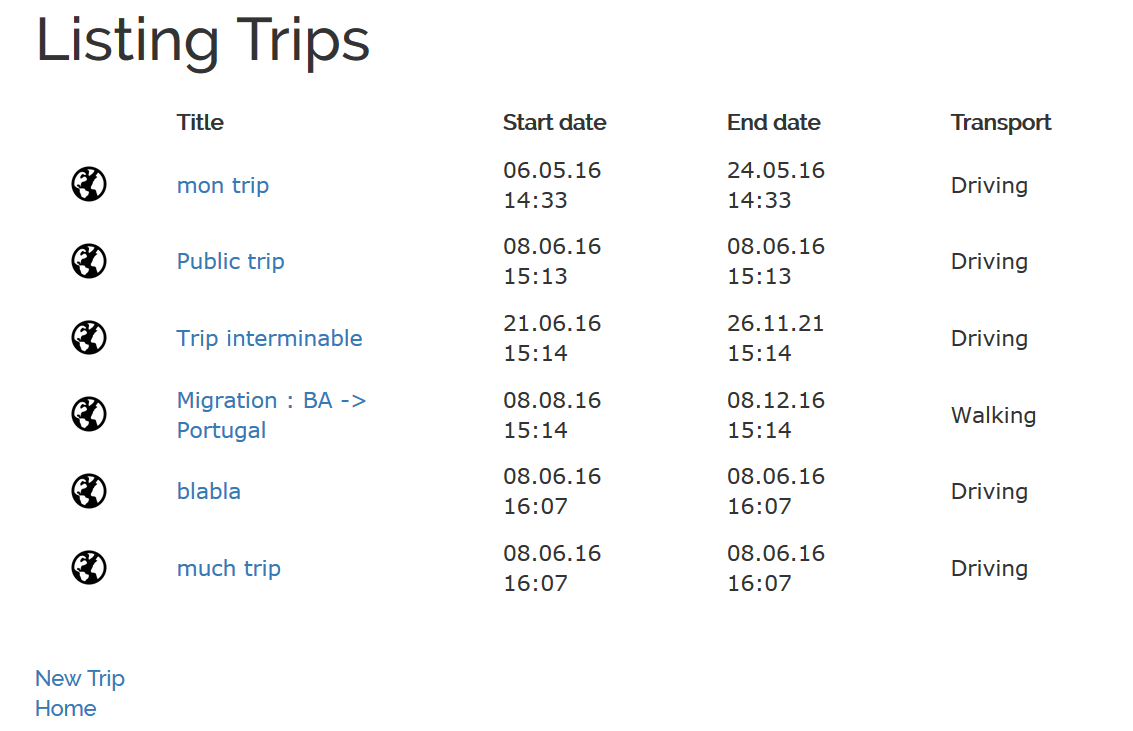


Une fois son compte créé, il est automatiquement connecté :

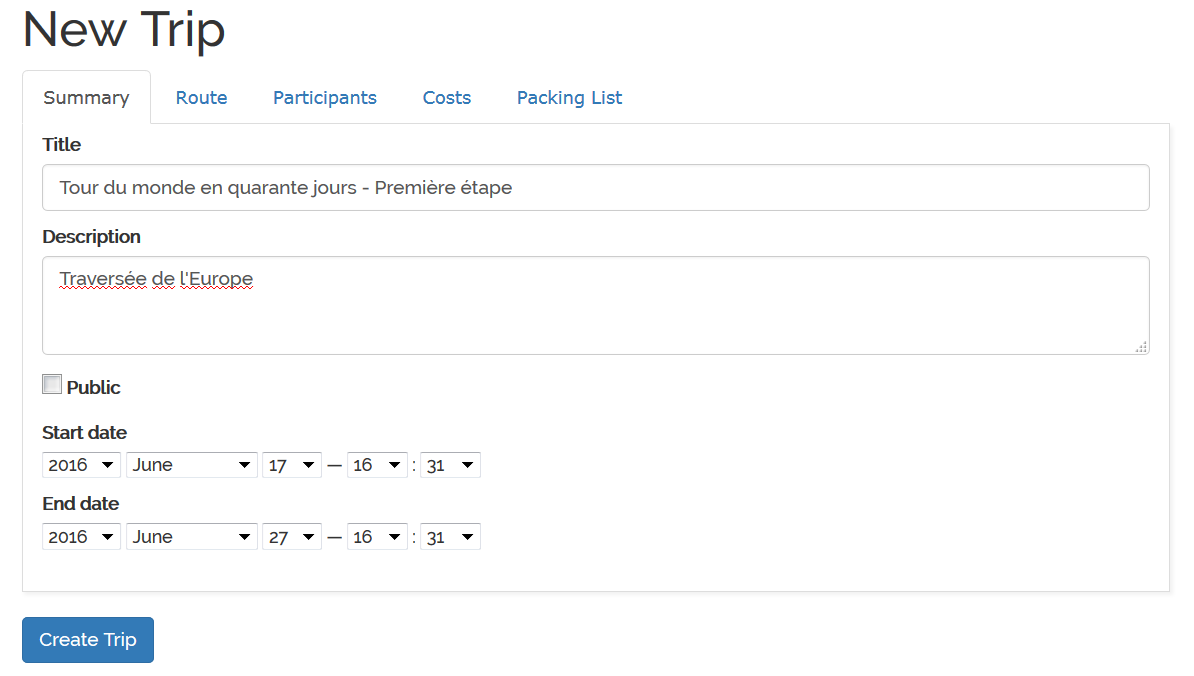


En cliquant sur « My Account », il a accès au même formulaire que lors de son inscription mais pour effectuer cette fois des modifications à son compte. Il peut se déconnecter. Pour finir il peut voir si il a reçu des invitations à rejoindre un trip. En l’occurrence, il vient de créer son compte donc personne n’a eu le temps de l’inviter.

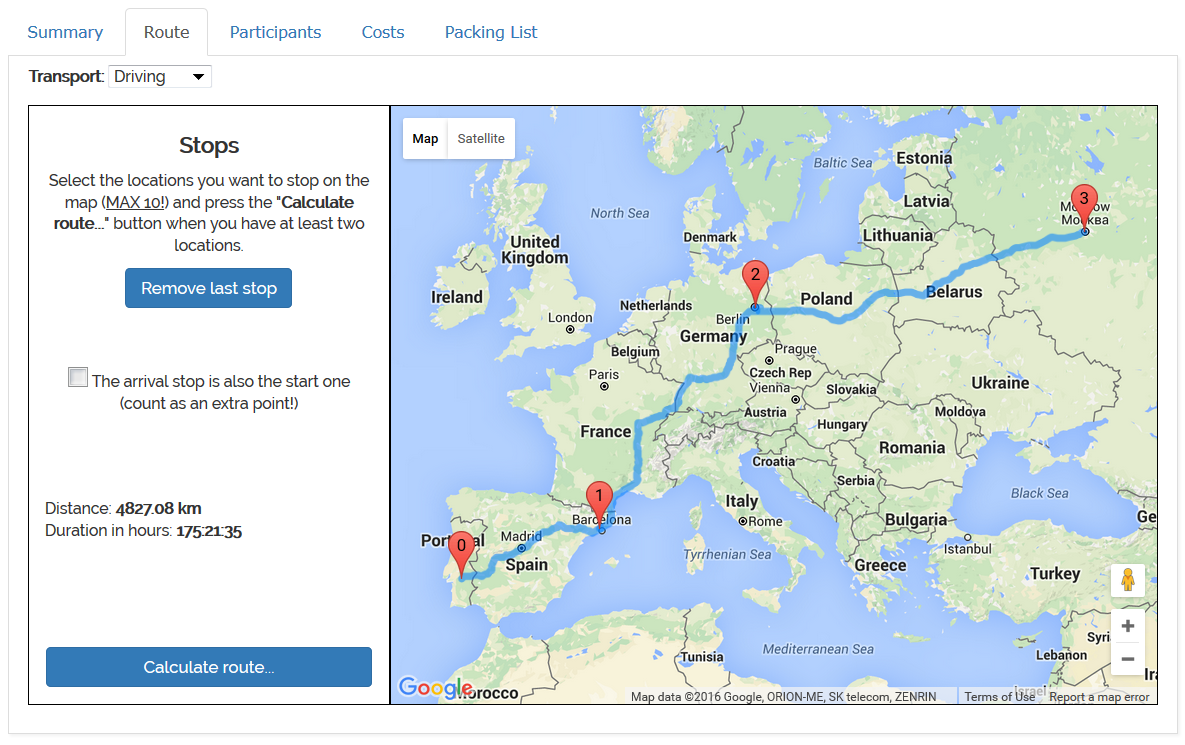
Il peut voir la totalité des trips publics ainsi que ceux dont il fait partie. Vu qu’il ne fait pas encore partie d’un trip, il voit seulement les trips publics.



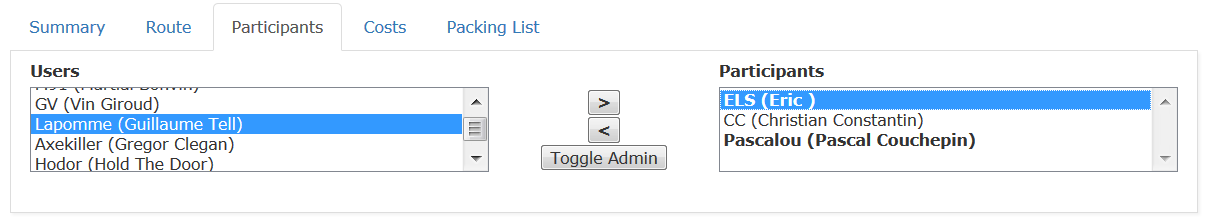
En cliquant sur « New Trip », il peut organiser son propre trip. Tout d’abord il doit fournir des informations générales :

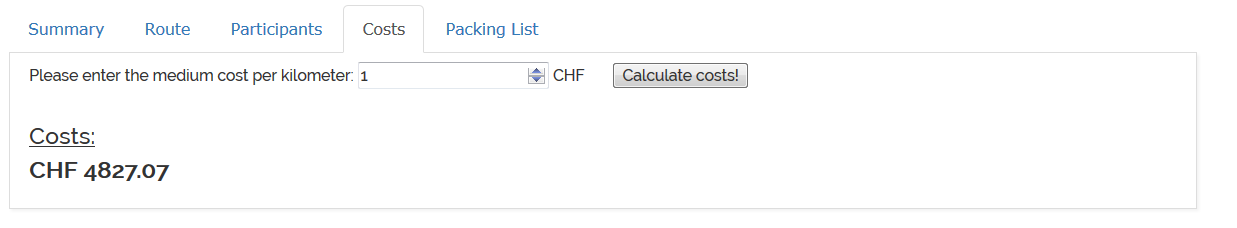


Ensuite mentionner une route et un moyen de transport. Pour choisir une choisir, il doit placer un point sur la carte pour chaque étape puis cliquer sur « Calculate Route » pour déterminer le temps et les kilomètres à effectuer. Dans l’exemple si dessous, il choisit de partir du Portugal et fait étape à Barcelone, Berlin puis arrive à Moscou.



Il peut ensuite inviter deux de ses amis : Christian et Pascal. Il fait confiance à Couchepin, il peut donc lui donner les droits d’admin grâce à « Toggle Admin »



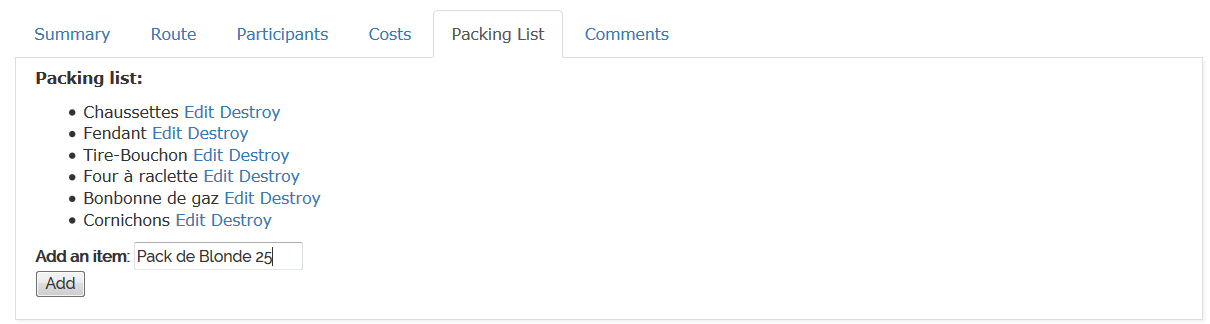
Il peut obtenir une approximation de coût pour son voyage en fournissant un montant par kilomètre : 

Il peut maintenant confirmer la création du trip grâce au bouton « Create Trip ».

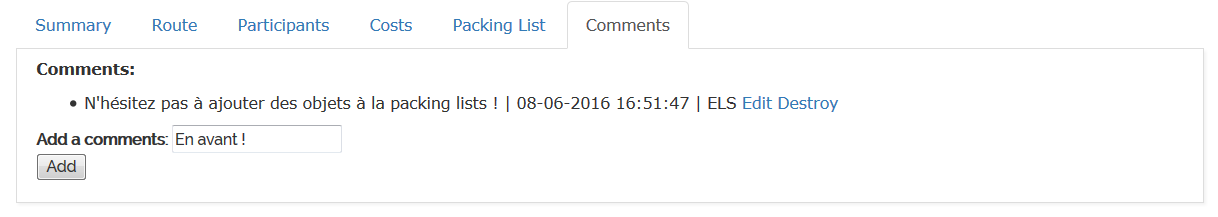
Il apparaît désormais dans la liste des voyages :



En cliquant sur le nom du trip, il peut le consulter et ainsi ajouter des objets à ne pas oublier en préparant son départ :

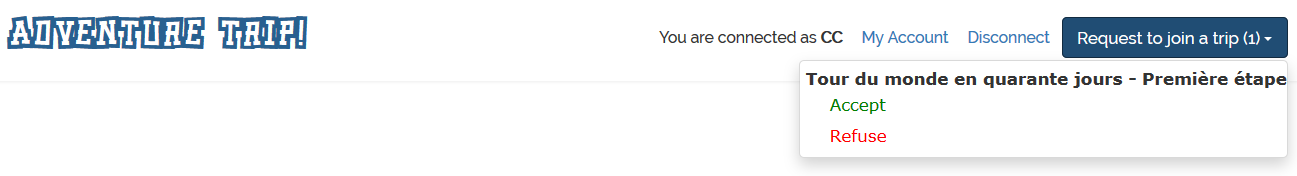


Ou ajouter un commentaire destiné aux autres participants.



Il peut également modifier les informations relatives aux trips (descriptions, itinéraire, participant) grâce au bouton Edit en bas de page. L’édition est similaire à la création.

Christian a reçu une invitation :



En acceptant, il pourra consulter les détails du trips mais sans pouvoir l’éditer car il n’a pas été nommé admin :



Il peut néanmoins écrire un message ou ajouter un objet à la packing list :

