

# *Find your ride*

CFPT en informatique

Technicien ES en informatique

Travail de diplôme



Classe : T.IS-E2A

Session : 2016-2017

Elève :  
Lucien Camuglia

Enseignant :  
M. Zeltner

9 juin 2017

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Résumé . . . . .	5
1.2	Abstract . . . . .	5
<b>2</b>	<b>Cahier des charges</b>	<b>6</b>
2.1	Description . . . . .	6
2.2	But . . . . .	6
2.3	Spécifications . . . . .	6
2.4	Restrictions . . . . .	6
2.5	Environnement . . . . .	6
2.6	Livrables . . . . .	6
2.7	Calendrier . . . . .	6
<b>3</b>	<b>Analyse de l'existant</b>	<b>7</b>
3.1	Garmin BaseCamp . . . . .	7
3.2	<a href="http://www.calculitineraires.fr/">http://www.calculitineraires.fr/</a> . . . . .	7
3.3	<a href="http://www.bestbikingroads.com">http://www.bestbikingroads.com</a> . . . . .	7
3.4	Conclusion . . . . .	7
<b>4</b>	<b>Analyse fonctionnelle</b>	<b>8</b>
4.1	Généralités . . . . .	8
4.2	Description des fonctionnalités globales . . . . .	9
4.2.1	Authentification . . . . .	9
4.2.2	Inscription . . . . .	9
4.2.3	Création de trajet . . . . .	9
4.2.4	Exportation . . . . .	9
4.2.5	Suppression d'un trajet . . . . .	9
4.2.6	Visualiser le trajet . . . . .	9
4.2.7	Contact . . . . .	9
4.3	Description de l'interface . . . . .	10
4.3.1	Inscription . . . . .	10
4.3.2	Map . . . . .	10
4.3.3	Detail trajet . . . . .	11
4.3.4	Connexion . . . . .	11

---

4.3.5	Vos trajets . . . . .	12
4.4	Description des éléments de sécurité . . . . .	12
4.4.1	Fichier .htaccess . . . . .	12
4.4.2	Utilisateur de la base de données . . . . .	12
4.4.3	Requêtes . . . . .	12
<b>5</b>	<b>Analyse organique</b>	<b>13</b>
5.1	Résumé . . . . .	13
5.2	<i>Bootstrap</i> . . . . .	13
5.3	Google API . . . . .	13
5.3.1	Google Maps Javascript . . . . .	13
5.3.2	Google Direction . . . . .	13
5.3.3	Google Elevation . . . . .	14
5.3.4	Google Roads . . . . .	14
5.4	Fichier GPX . . . . .	14
5.5	Fonction PHP pour la base de données . . . . .	15
5.5.1	connexion à la base de données . . . . .	15
5.5.2	Requêtes préparées . . . . .	15
5.5.3	GetMotorcycleBrand . . . . .	15
5.5.4	GetMotorcycleModel . . . . .	15
5.5.5	GetAllMotorcycleYear . . . . .	15
5.5.6	GetAllMotorcycleConsumption . . . . .	15
5.5.7	GetAllMotorcycleTiredness . . . . .	16
5.5.8	GetAllMotorcycles . . . . .	16
5.5.9	GetRoutes . . . . .	16
5.5.10	CreateRoute . . . . .	16
5.5.11	deletePlaces . . . . .	16
5.5.12	addPlaceToRoute . . . . .	16
5.5.13	deleteMotorcycle . . . . .	17
5.5.14	getUserNMotorcycle . . . . .	17
5.5.15	getUserRoleById . . . . .	17
5.5.16	AddSinuosity . . . . .	17
5.5.17	AddElevation . . . . .	17
5.5.18	AddLength . . . . .	17
5.5.19	GetmostSinuousRoad . . . . .	17

5.5.20	GetLessSinuousRoad . . . . .	18
5.5.21	GetMostSteepestRoad . . . . .	18
5.5.22	GetLessSteepestRoad . . . . .	18
5.6	Inscription sur le site . . . . .	19
5.7	Connexion au site . . . . .	20
5.8	Upload d'un itinéraire . . . . .	22
5.9	Téléchargement d'un itinéraire . . . . .	23
5.10	Fonction javascript de gestion de la carte . . . . .	24
5.10.1	Initialisation de la carte . . . . .	24
5.10.2	ImportGPX . . . . .	27
5.10.3	AskGoogle . . . . .	27
5.10.4	LoadPoints . . . . .	27
5.10.5	SnappPoints2Road . . . . .	27
5.10.6	SaveNewLocation . . . . .	27
5.10.7	CreateRoute . . . . .	28
5.10.8	RouteClick . . . . .	28
5.10.9	searchIti . . . . .	29
5.10.10	Clear . . . . .	29
5.10.11	ShowParcours . . . . .	29
5.10.12	DisplayRoute . . . . .	30
5.10.13	CreateMarker . . . . .	30
5.10.14	refreshValues . . . . .	30
5.10.15	DownloadRoute . . . . .	30
5.10.16	EnableDisabledFilters . . . . .	30
5.10.17	RefreshRouteWithFilters . . . . .	31
5.10.18	RefreshRouteWithoutFilters . . . . .	31
5.10.19	StartCreation . . . . .	31
5.10.20	DisplayRouteInfo . . . . .	31
5.10.21	DeleteRoute . . . . .	31
5.10.22	ClearRoute . . . . .	32
5.10.23	Document.Ready . . . . .	32
5.10.24	Document.bind . . . . .	32
5.11	fonction javascript pour l'inscription et la connexion . . . . .	32
5.11.1	connexion . . . . .	32
5.11.2	LoadModalFromBrand . . . . .	32

5.11.3	LoadYearFromModel . . . . .	33
5.11.4	UserNameExists . . . . .	33
5.11.5	checkForEnabled . . . . .	33
5.11.6	Document.Ready . . . . .	33
5.12	Fonction ajax . . . . .	33
5.12.1	GetModel . . . . .	33
5.12.2	GetYear . . . . .	33
5.12.3	UserExists . . . . .	34
5.12.4	GetMotorcycles . . . . .	34
5.12.5	GetRoutePoints . . . . .	34
5.12.6	SaveNewRoute . . . . .	34
5.12.7	AddMotorcycle . . . . .	34
5.12.8	UpdateUserRole . . . . .	34
5.12.9	GetUserRole . . . . .	35
5.12.10	downloadRoute . . . . .	35
5.12.11	FilterRoad . . . . .	35
5.12.12	GetRoutesJSON . . . . .	35
5.12.13	GetRoadsInfos . . . . .	35
5.12.14	CreateNewRoute . . . . .	35
5.12.15	DeleteRoute . . . . .	36
5.13	Base de données . . . . .	37
5.13.1	Modèle conceptuel . . . . .	37
5.13.2	Structure de la base de données . . . . .	37
<b>6</b>	<b>Estimation de l'apport personnel</b>	<b>39</b>
<b>7</b>	<b>Problèmes rencontré</b>	<b>40</b>
7.1	Google Api . . . . .	40
7.2	Asynchrone . . . . .	40
<b>8</b>	<b>Conclusion</b>	<b>41</b>
8.1	Bilan personnel . . . . .	41
8.2	Améliorations . . . . .	41

# 1 Introduction

## 1.1 Résumé

Site WEB permettant le partage d'itinéraire entre passionné de la moto. Ce site est réalisé en HTML/PHP/CSS/AJAX et incluant les API<sup>1</sup> Google.

Il permet à un utilisateur de créer un trajet, de le partager et de le modifier.

Un utilisateur peut rechercher un itinéraire avec différents critères :

- Durée
- Type de route
- Changement d'altitude

Le site fournit à l'utilisateur différentes données comme le temps du trajet ou la consommation théorique de la moto pour la balade.

Le motard peut aussi importer et exporter des fichiers directement depuis son GPS.

## 1.2 Abstract

The website allows sharing itineraries between motorcycle enthusiasts. The site is set-up with HTML/PHP/CSS/AJAX and Google's API. It allows users to create trips, to share and modify them.

Different criteria can be defined by the user to create the itinerary :

- Travel Time
- Type of road
- Change in altitude, etc.

This website gives the user diverse informations. One example could be the theoretical consumption of their motorcycle for the ride or travel time. The rider can also upload or download files directly from his GPS.

---

1. *Application Programming Interface*(Interface de Programation Applicative) [https://fr.wikipedia.org/wiki/Interface\\_de\\_programmation](https://fr.wikipedia.org/wiki/Interface_de_programmation)

## 2 Cahier des charges

### 2.1 Description

Site web permettant aux motard de partager leurs excursions.

### 2.2 But

Le but de ce travail est d'avoir un site web permettant le partage et la communication entre les passionnés de la moto.

Le site permettra d'afficher sur une carte le tracé du trajet, la durée, leur appréciation personnelle et celle de la communauté, divers commentaires et éventuellement la consommation d'essence.

Le site donnera aussi la possibilité au motard d'importer ou d'exporter des tracés depuis son GPS à l'aide de fichier .GPX

Le tout sera développé autour de l'API Google Map.

### 2.3 Spécifications

- Site web réalisé en HTML/CSS/PHP/JS/AJAX
- Base de données SQL
- API Google map

### 2.4 Restrictions

- Responsive design pour la consultation sur smartphone/tablettes

### 2.5 Environnement

- PC standard de l'école avec droit d'administrateur
- Netbeans IDE 7.4
- EasyPHP Devserver 7.1.1

### 2.6 Livrables

- Poster
- Dossier d'analyse
- CDROM avec les sources
- Journal de bord

### 2.7 Calendrier

- début : mer 05/04/17
- reddition intermédiaire (doc + poster) : ven 06/05/17
- reddition finale : lun 12/06/17

## 3 Analyse de l'existant

### 3.1 Garmin BaseCamp

Garmin BaseCamp est un logiciel fournit par Garmin.

Il permet à un utilisateur de créer un trajet et de l'importer sur son GPS.

Il donne aussi la possibilités a l'utilisateur de visionner ses différents déplacements.

Positif	Négatif
<ul style="list-style-type: none"><li>• Connexion directe avec le GPS</li></ul>	<ul style="list-style-type: none"><li>• Pas de partage</li><li>• Obligation de posséder un GPS Garmin</li></ul>

### 3.2 <http://www.calculitineraires.fr/>

[www.calculitineraires.fr](http://www.calculitineraires.fr) est un site de partage d'itinéraire pour la course à pied, le vélo et la randonnée

Il permet l'import/export de fichier GPX et TCX, la rechercher et le partage d'itinéraire.

Positif	Négatif
<ul style="list-style-type: none"><li>• Assez complet</li><li>• Import / Export des fichiers GPS</li><li>• Création d'itinéraire</li></ul>	<ul style="list-style-type: none"><li>• Interface compliquée d'utilisation</li><li>• Pas de contribution pour la moto</li></ul>

### 3.3 <http://www.bestbikingroads.com>

[www.bestbikingroads.com](http://www.bestbikingroads.com) est un site de partage d'itinéraire moto.

Il permet l'import/export de fichier GPX, la rechercher, le partage d'itinéraire ainsi que la notation des balades.

Positif	Négatif
<ul style="list-style-type: none"><li>• Assez complet</li><li>• Import / Export des fichiers GPS</li><li>• Création d'itinéraire</li><li>• Notation des balades</li></ul>	<ul style="list-style-type: none"><li>• Tout les tracés s'affiche en meme temps sur la carte</li><li>• pas de modification possible</li></ul>

### 3.4 Conclusion

Il existe différent site de partage d'itinéraire, cependant ils ont tous des fonctionnalités assez similaire. Mon site se distingue des autres de part son interface intuitive pour l'utilisateur et certaines fonctionnalités trouvée sur aucun site.

Par exemple il est possible sur *Find your ride* de calculer sa consommation théorique pour l'itinéraire ou de filtrer les routes selon leurs sinuosité.



## 4 Analyse fonctionnelle

### 4.1 Généralités

Ci-dessous se trouve le schéma initial de mon site web. Les ronds représentent des pages et les flèches entre ceux-ci représentent d'éventuelles actions ou états.

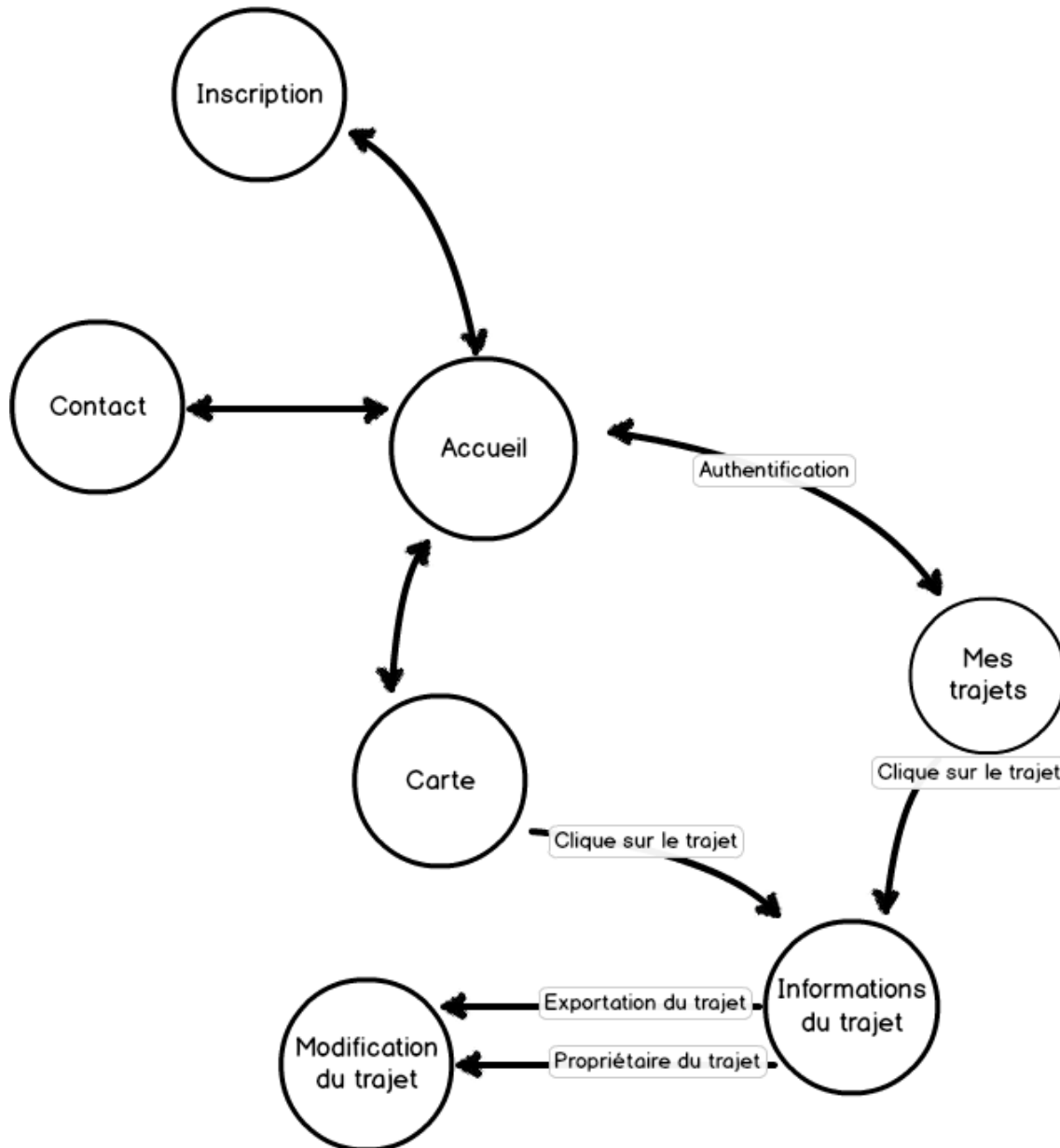


FIGURE 1 – Schéma du site

## 4.2 Description des fonctionnalités globales

### 4.2.1 Authentification

Cette fonctionnalité permet à un utilisateur de s'authentifier et d'accéder à ses trajets mit en ligne ou partager de nouveaux trajets.

### 4.2.2 Inscription

Cette fonctionnalité permet à un nouvel utilisateur de créer un compte et donc de pouvoir bénéficier des fonctionnalités d'un utilisateur connecté

### 4.2.3 Création de trajet

Cette fonctionnalité se distingue en deux sous fonctionnalités :

- Création : crée un nouveau trajet depuis le site directement.
- Importation : importe un fichier de type GPX et éventuellement modifier le trajet.

### 4.2.4 Exportation

Cette fonctionnalité permet à un utilisateur d'exporter le trajet de son choix au format GPX pour l'inclure dans son GPS.

### 4.2.5 Suppression d'un trajet

Cette fonction permet a un utilisateur de supprimer un de ses trajet.

### 4.2.6 Visualiser le trajet

Cette fonctionnalité permet à n'importe qui de visualiser les trajets mit en ligne, puis les exporter en les modifiant s'il le souhaite.

### 4.2.7 Contact

Cette fonctionnalité permet à n'importe qui de contacter le webmaster afin de faire des remarques ou des demandes.

## 4.3 Description de l'interface

### 4.3.1 Inscription

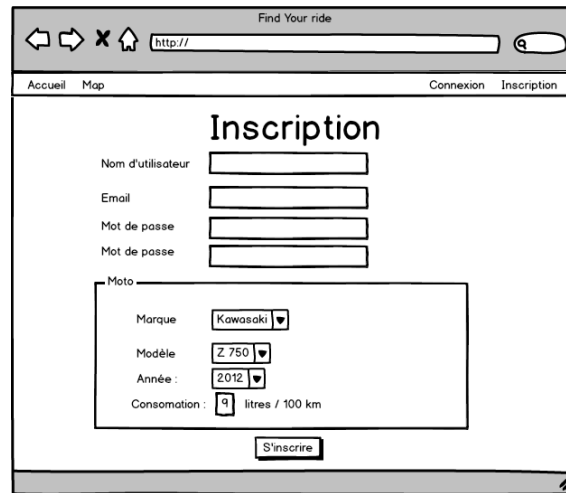


FIGURE 2 – Page inscription

Cette page sert à l'inscription des utilisateurs. On y trouve différents champs :

- Nom d'utilisateur
- E-mail
- Mot de passe, ce champ apparaît deux fois pour avoir la validation de celui-ci
- Moto : diverses informations sur la moto de l'utilisateur comme par exemple sa consommation pour calculer la consommation des trajets.

### 4.3.2 Map

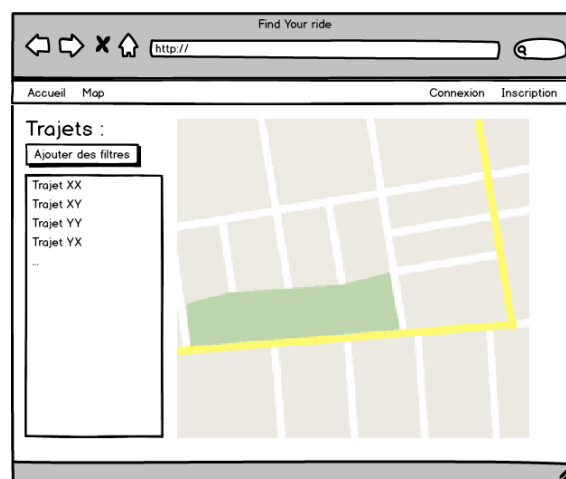


FIGURE 3 – Page inscription

Cette page sert à l'affichage de la carte et des différents trajets.

Sur la gauche apparaissent tous les trajets ainsi qu'un bouton filtre. Ce bouton permet de filtrer les trajets parmi différents critères.

- Avec ou sans autoroute

- Durée
- Dénivelé

Sur la droite une carte Google ou apparaît le tracé sélectionné.

#### 4.3.3 Detail trajet

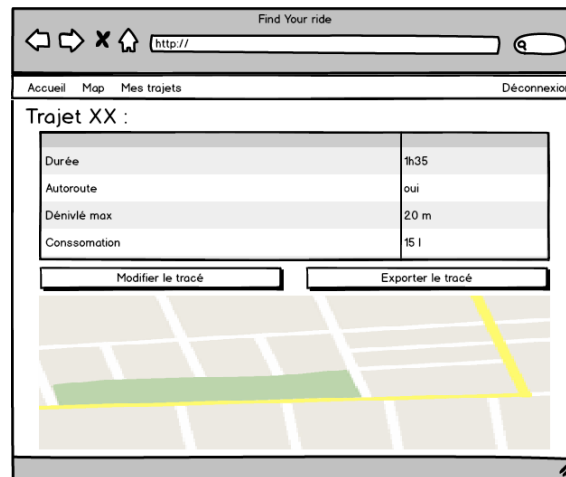


FIGURE 4 – Page Detail Trajet

Cette page sert à afficher le détail d'un trajet.

Sur le haut de la page apparai les détails du trajet.

- Durée du trajet
- S'il contient des autoroutes
- Dénivelé
- La consommation (si l'utilisateur à renseignée les données de la moto)

Deux bouton sont présent pour modifier ou exporter le trajet. Sur le bas de la page, la carte google avec le trajet.

#### 4.3.4 Connexion

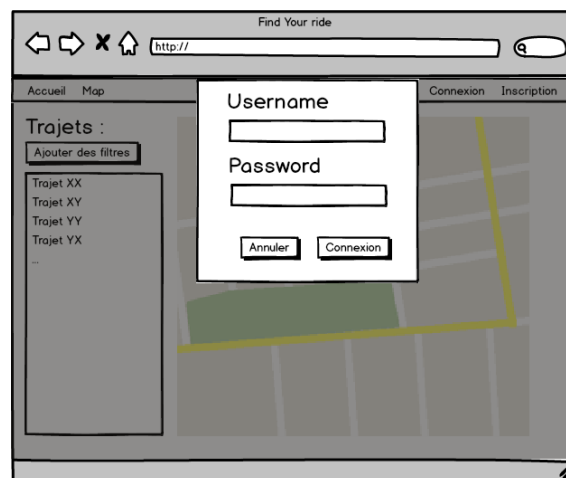


FIGURE 5 – Modal connexion

Le formulaire de connexion est une fenêtre modal qui s'ouvre par dessus les autres pages avec uniquement deux champs.

- Nom d'utilisateur
- Mot de passe

#### 4.3.5 Vos trajets

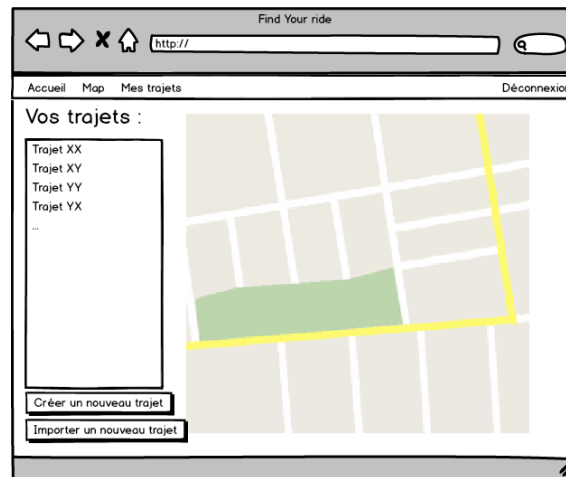


FIGURE 6 – Page vos trajet

Cette page est identique que la page *Map* mais au lieu d'avoir tout les trajet disponible du site, il y a uniquement les trajet de l'utilisateur.

Deux boutons sont disponible sur cette page.

- Créer un nouveau trajet, permet de créer un trajet a partir de la carte
- Importer un nouveau trajet, permet d'importer un fichier GPX avec le trajet.

## 4.4 Description des éléments de sécurité

### 4.4.1 Fichier .htaccess

Permet d'empêcher la navigation sur certain dossier/fichiers du site. Ils permettent aussi de définir des pages d'erreurs personnalisées

### 4.4.2 Utilisateur de la base de données

Utiliser un utilisateur différent que *root* pour accéder à la base de donnée afin de donner des droits qu'au éléments nécessaire.

### 4.4.3 Requêtes

Utilisation de requêtes préparée pour éviter les injections SQL.

## 5 Analyse organique

### 5.1 Résumé

Dans cette section sera expliqué le fonctionnement de l'application.

### 5.2 *Bootstrap*

N'étant pas designer web j'ai décidé de réaliser le design du site à l'aide du framework<sup>2</sup> *bootstrap*.

Le framework *bootstrap* initialement appelé *Twitter Blueprint* à été créé en 2010 par *Twitter*.

Il offre une collection d'outils d'aide à la création du design d'un site web.

L'avantage de cet outil est la compatibilité avec les versions récentes des navigateurs, il est aussi *responsive* ce qui permet la consultation du site sur tout types d'écran, que ce soit ordinateur, tablette ou smartphone.

### 5.3 Google API

Les API<sup>3</sup> Google sont une suite d'outils développée et fournie par Google pour permettre à des utilisateurs d'intégrer les services Google dans un site web ou une application.

Il en existe environ 66 dont 16 pour *Google Maps*. Pour la réalisation de ce site j'utilise 4 API *Google Maps* :

- Google Maps Javascript
- Google Direction
- Google Elevation
- Google Roads

#### 5.3.1 Google Maps Javascript

Cette API permet d'intégrer une *Google Maps* sur le site web, c'est la seule qui sera visible pour l'utilisateur. C'est sur celle-ci qu'apparaissent les itinéraires et les positions GPS.

#### 5.3.2 Google Direction

Le fonctionnement de cette API est assez simple. Il suffit d'envoyer un point d'origine et une destination à Google Direction et il nous retourne un tableau JSON avec différents paramètres.

- La durée
- La distance
- Les étapes du parcours (positions GPS)
- Les étapes du parcours (Français, par exemple : Tournez à droite,...)

---

2. Infrastructure de développement. <https://fr.wikipedia.org/wiki/Framework>

3. *Application Programming Interface* (Interface de Programmation Applicative) [https://fr.wikipedia.org/wiki/Interface\\_de\\_programmation](https://fr.wikipedia.org/wiki/Interface_de_programmation)

### 5.3.3 Google Elevation

Cette API permet d'obtenir l'altitude de points géographique à la surface de la Terre, y compris les fonds marins.

Les données géographique peuvent être fournies de deux façons différentes :

- En tant que point géographique
- En tant que tracé

Pour mon site l'utilisation sera faite à l'aide de tracé afin d'avoir les altitudes de chaque point de la route.

### 5.3.4 Google Roads

Cette API permet de faire 3 choses différentes :

- *Snap to road*
- *Nearest road*
- *Speed limits*

J'utilise deux fonctionnalités de cette API, *Snap to road* et *Nearest road*.

*Snap to road* permet de "déplacer" un trajet pour qu'il suive la route. Il faut lui envoyer l'ensemble des points GPS de notre route et il nous retourne un ensemble de points qui suivent la route.

*Nearest road* permet de donner la position d'un point GPS sur la route la plus proche.

## 5.4 Fichier GPX

GPX est un format de fichier ouvert permettant l'échange de coordonnées GPS. Il peut contenir trois types d'informations :

- Point de cheminement (*waypoint*)
- Route
- Trace

Le fichier suit la même structure que le XML.

Ci-dessous, un exemple de fichier GPX

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <gpx version="1.0" creator="FindYourRide.org" xmlns:xsi="http://www
  .w3.org/2001/XMLSchema-instance" xmlns="http://www.topografix.
  com/GPX/1/0" xsi:schemaLocation="http://www.topografix.com/GPX
  /1/0 http://www.topografix.com/GPX/1/0/gpx.xsd">
3
4 <trk>
5 <name>Journal actif: 2016-06-28 15:47</name>
6 <trkseg>
7 <trkpt lat="46.188097" lon="6.196411">
8 <ele>397.64</ele>
9 <time>2016-06-28T14:47:33Z</time>
10 <extensions>
11 <gpxtpx:TrackPointExtension>
12 <gpxtpx:course>0.00</gpxtpx:course>
13 </gpxtpx:TrackPointExtension>
14 </extensions>
15 </trkseg>
16 </trkpt>
17 </trk>
```

## 5.5 Fonction PHP pour la base de données

### 5.5.1 connexion à la base de données

La connexion à la base de données se fait à l'aide de PDO<sup>4</sup>. PDO a besoin de l'utilisateur de la base de données et du mot de passe ainsi que le nom de la base. Je lui précise aussi le mode d'erreur qui est `PDO::ERRMODE_EXCEPTION` ce mode permet d'afficher le code d'erreur et de déclencher une exception<sup>5</sup>.

Afin d'avoir un partage d'informations dans le bon format, on définit l'encodage de caractères en UTF-8

### 5.5.2 Requêtes préparées

Afin de sécuriser le site et éviter les injections SQL j'utilise des requêtes préparées.

On commence par préparer la requête, PDO va substituer les marqueurs ( `:marqueur`) pour les valeurs fournies dans le tableau de paramètres fourni au moment de l'exécution. Ensuite, il faut exécuter la requête avec les paramètres.

### 5.5.3 GetMotorcycleBrand

Ne prend pas de paramètre d'entrée.

Récupère dans la base de données toutes les marques de moto.

Retourne un tableau contenant les marques.

### 5.5.4 GetMotorcycleModel

Ne prend pas de paramètre d'entrée.

Récupère dans la base de données tous les modèles de moto.

Retourne un tableau contenant les modèles triés par ordre alphabétique.

### 5.5.5 GetAllMotorcycleYear

Ne prend pas de paramètre d'entrée.

Récupère dans la base de données toutes les années.

Retourne un tableau contenant les années dans l'ordre croissant.

### 5.5.6 GetAllMotorcycleConsumption

Ne prend pas de paramètre d'entrée.

Récupère dans la base de données toutes les consommations.

---

4. *PHP Data Object*, <http://php.net/manual/fr/intro.pdo.php>

5. Plus d'informations <http://php.net/manual/fr/pdo.error-handling.php>



Retourne un tableau contenant les consommations dans l'ordre croissant.

#### **5.5.7 GetAllMotorcycleTiredness**

Ne prend pas de paramètre d'entrée.

Récupère dans la base de données tous les indices de fatigue.

Retourne un tableau contenant les indices de fatigue dans l'ordre croissant.

#### **5.5.8 GetAllMotorcycles**

Ne prend pas de paramètre d'entrée.

Récupère dans la base de données toutes les motos.

Retourne un tableau contenant toutes les motos et leurs informations trié par marque.

#### **5.5.9 GetRoutes**

Prend un paramètre facultatif qui est l'identifiant de l'utilisateur.

Récupère dans la base de données les routes. Soit toutes les routes, soit les route de l'utilisateur.

Retourne un tableau contenant les routes.

#### **5.5.10 CreateRoute**

Prend 3 données en entré, le nom de la route, l'identifiant de l'utilisateur et si la route contient des autoroutes.

Insère les données dans la base de données

Retourne l'identifiant de la route créée.

#### **5.5.11 deletePlaces**

Prend entrée l'identifiant de la route.

Supprime tous les points GPS de la route dans la base de données.

Ne retourne rien

#### **5.5.12 addPlaceToRoute**

Prend en entrée 4 paramètres, la latitude, la longitude, la position du point dans le tracé et l'identifiant de la route.

Insère les informations dans la base de données.

Ne retourne rien

**5.5.13 deleteMotorcycle**

Prend en entrée l'identifiant de la moto

Supprime la moto

Ne retourne rien.

**5.5.14 getUserNMotorcycle**

Ne prend pas de paramètre d'entrée.

Récupère toutes les informations d'un utilisateur et de sa moto.

Retourne un tableau avec toutes les informations.

**5.5.15 getUserRoleById**

Prend en entrée l'identifiant de l'utilisateur.

Sélectionne dans la base de données le rôle de l'utilisateur.

Retourne le rôle.

**5.5.16 AddSinuosity**

Prend en entrée l'identifiant de la route et la sinuosité.

Ajoute la sinuosité dans la base de données

Ne retourne rien.

**5.5.17 AddElevation**

Prend en entrée l'identifiant de la route et l'altitude.

Ajoute l'altitude dans la base de données

Ne retourne rien.

**5.5.18 AddLength**

Prend en entrée l'identifiant de la route et la longueur.

Ajoute la longueur dans la base de données

Ne retourne rien

**5.5.19 GetmostSinuousRoad**

Ne prend rien en entrée.

Sélectionne dans la base de données la route la plus sinueuse.

Retourne la valeur de la route la plus sinueuse.

**5.5.20 GetLessSinuousRoad**

Ne prend rien en entrée.

Sélectionne dans la base de données la route la moins sinueuse.

Retourne la valeur de la route la moins sinueuse.

**5.5.21 GetMostSteepestRoad**

Ne prend rien en entrée.

Sélectionne dans la base de données la route la plus "pentue".

Retourne la valeur de la route la plus "pentue".

**5.5.22 GetLessSteepestRoad**

Ne prend rien en entrée.

Sélectionne dans la base de données la route la moins "pentue".

Retourne la valeur de la route la moins "pentue".

## 5.6 Inscription sur le site

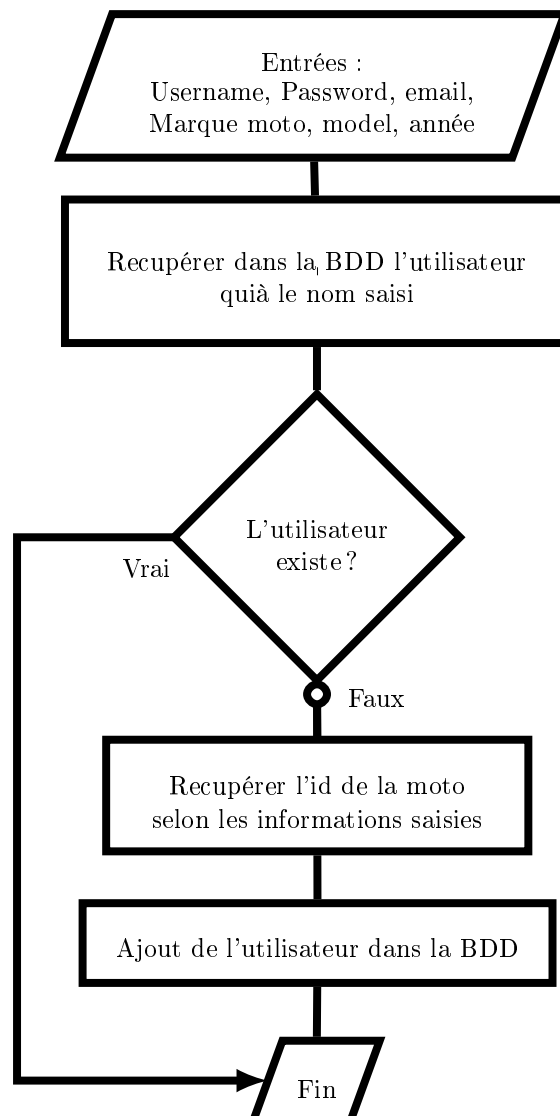
Afin de pouvoir bénéficier de toutes les fonctionnalités du site, l'utilisateur doit se connecter. Si celui-ci n'a pas de compte, il a la possibilité d'en créer un.

La création d'un utilisateur se fait en plusieurs étapes :

- Contrôle en AJAX lors de la saisie des informations par l'utilisateur
- Contrôle en PHP des informations
- Création de l'utilisateur en PHP et SQL

La vérification Ajax est assez simple, lorsque l'utilisateur appuie sur une touche, on affiche une croix rouge puis on envoie une requête à la page PHP qui va nous retourner un booléen à *true* si l'utilisateur existe. S'il est vrai, on laisse la croix rouge sinon on affiche un vu vers.

Une fois la vérification faite en javascript/ajax, les données sont envoyées à une fonction PHP qui vérifie si l'utilisateur est déjà présent dans la base, puis, récupère l'identifiant de la moto d'après la marque, le modèle et l'année. Finalement, les données sont ajoutées à la base de données.



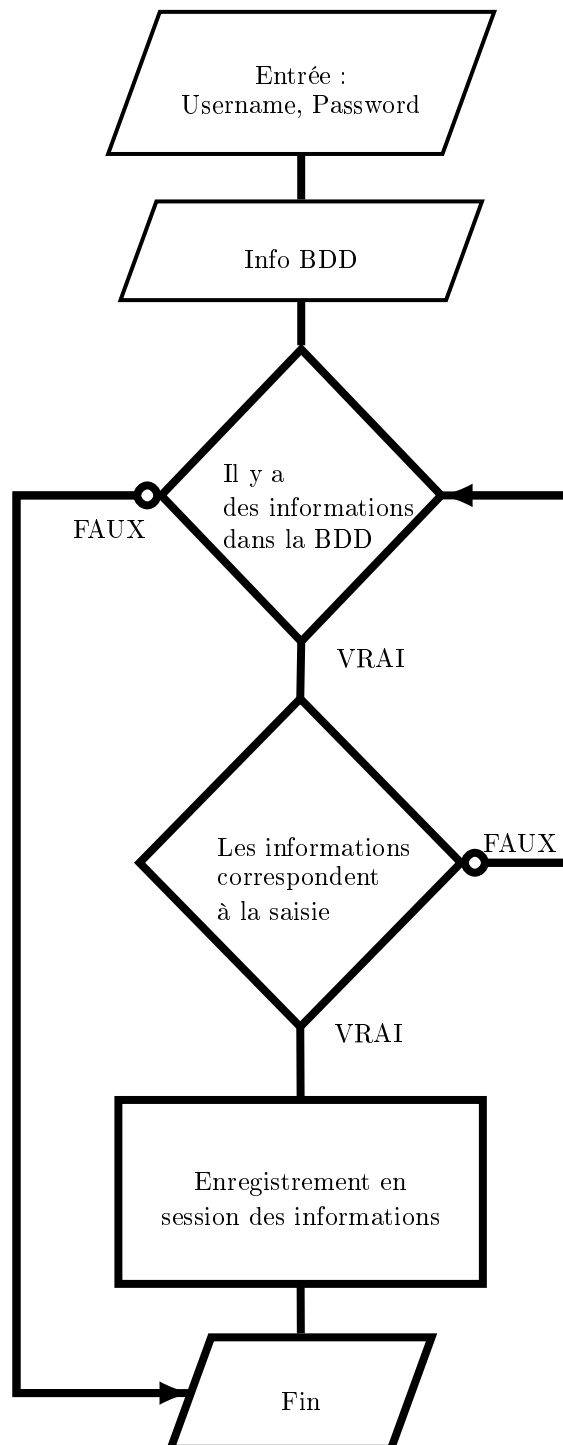
## 5.7 Connexion au site

Une fois l'inscription effectuée, l'utilisateur a la possibilité de se connecter au site.

Pour ce faire, l'utilisateur saisi ses informations dans le formulaire et se connecte. Une première vérification en HTML5 vérifie que les champs soient remplis(*required*). Une fois cette vérification effectuée, les informations sont envoyée à la page PHP *connexion.php* qui vérifie encore une fois que les champs soient rempli. Puis, envoie les données à la fonction PHP de connexion qui va récupérer :

- Les ids utilisateur
- Les mots de passe
- Les roles

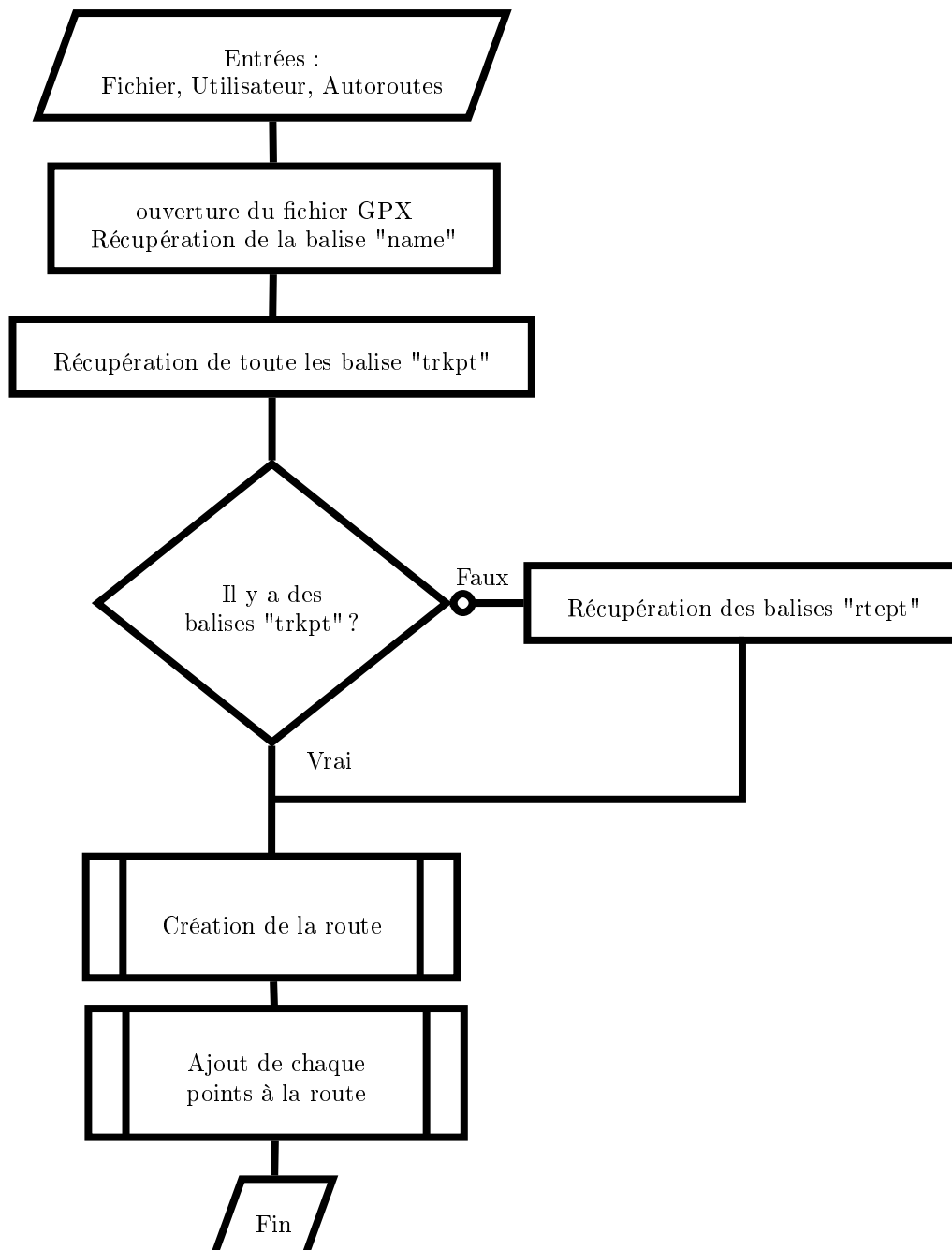
Une fois ces informations récupérée, on les parcourt tous pour être sûr que le nom d'utilisateur et le mot de passe fourni correspondent bien à un utilisateur existant. Si c'est le cas, on enregistre l'id, le role et le nom d'utilisateur dans des *Sessions* et on retourne *true* pour dire que les informations sont correcte.



## 5.8 Upload d'un itinéraire

L'utilisateur a la possibilité de mettre en ligne ses itinéraires depuis un fichier *.GPX*.

Premierement, un script vérifie l'extension du fichier. Une fois l'extension vérifiée, une fonction enregistre chaque point du fichier dans la base de données. Finalement, une fonction javascript envoie a google les point et récupère la position sur la route (API Google Roads) et enregistre les nouvelles coordonnées.



## 5.9 Téléchargement d'un itinéraire

Un itinéraire disponible sur le site est téléchargeable. Pour ce faire, l'utilisateur choisi l'itinéraire et clique sur *Download route*.

Le traitement se fait en PHP.

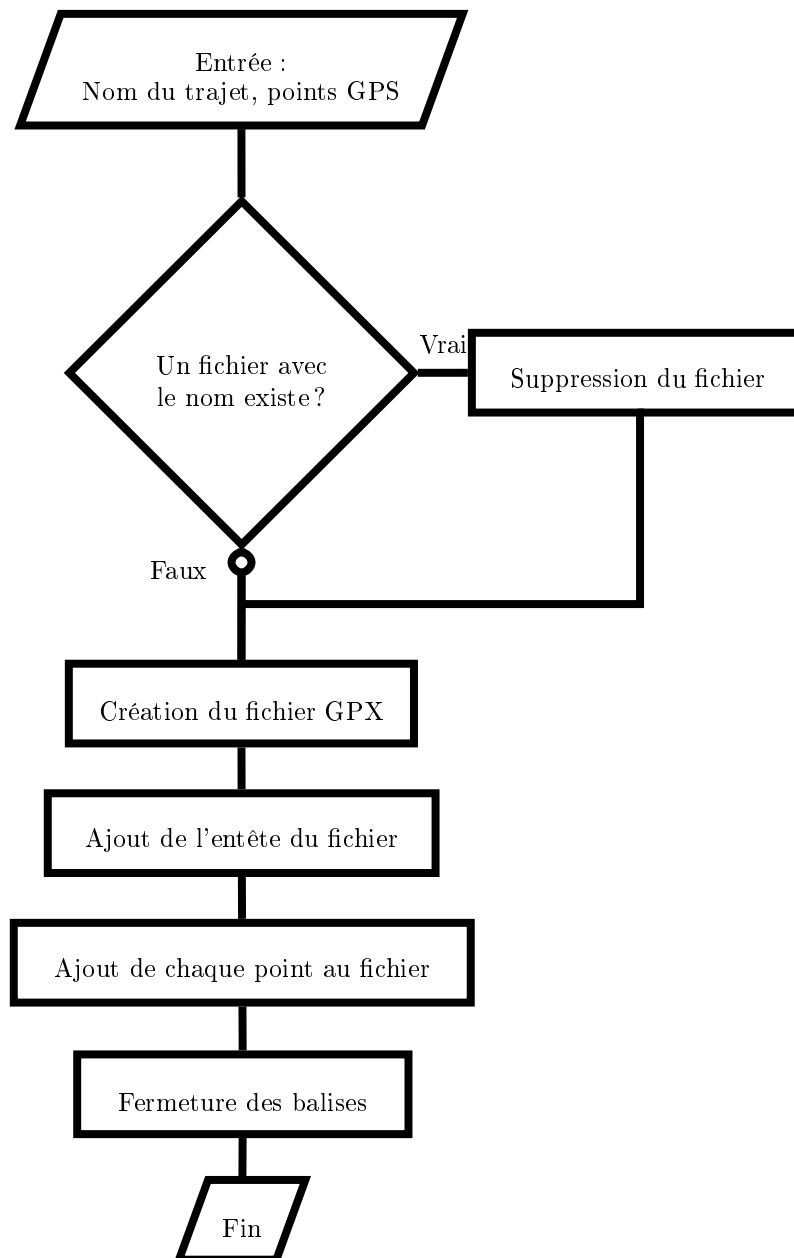
Le trajet et le nom sont transmit à la fonction qui va vérifier si un fichier du même nom existe déjà. Si c'est le cas, le fichier est supprimé et un autre est créé.

On ajoute ensuite l'entête du fichier XML et GPX.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <gpx version="1.0" creator="FindYourRide.org" xmlns:xsi="http://www
  .w3.org/2001/XMLSchema-instance" xmlns="http://www.topografix.
  com/GPX/1/0" xsi:schemaLocation="http://www.topografix.com/GPX
  /1/0 http://www.topografix.com/GPX/1/0/gpx.xsd">
```

Une fois ces informations insérée dans le fichier il faut ajouter le nom, chaque point et fermer les balises.





## 5.10 Fonction javascript de gestion de la carte

Ci-dessous sont toutes les fonctions utilisées pour créer et gérer la carte Google.

### 5.10.1 Initialisation de la carte

L'initialisation de la carte se fait à plusieurs moments :

- Chargement de la page
- Chargement d'un itinéraire
- Modification d'un itinéraire

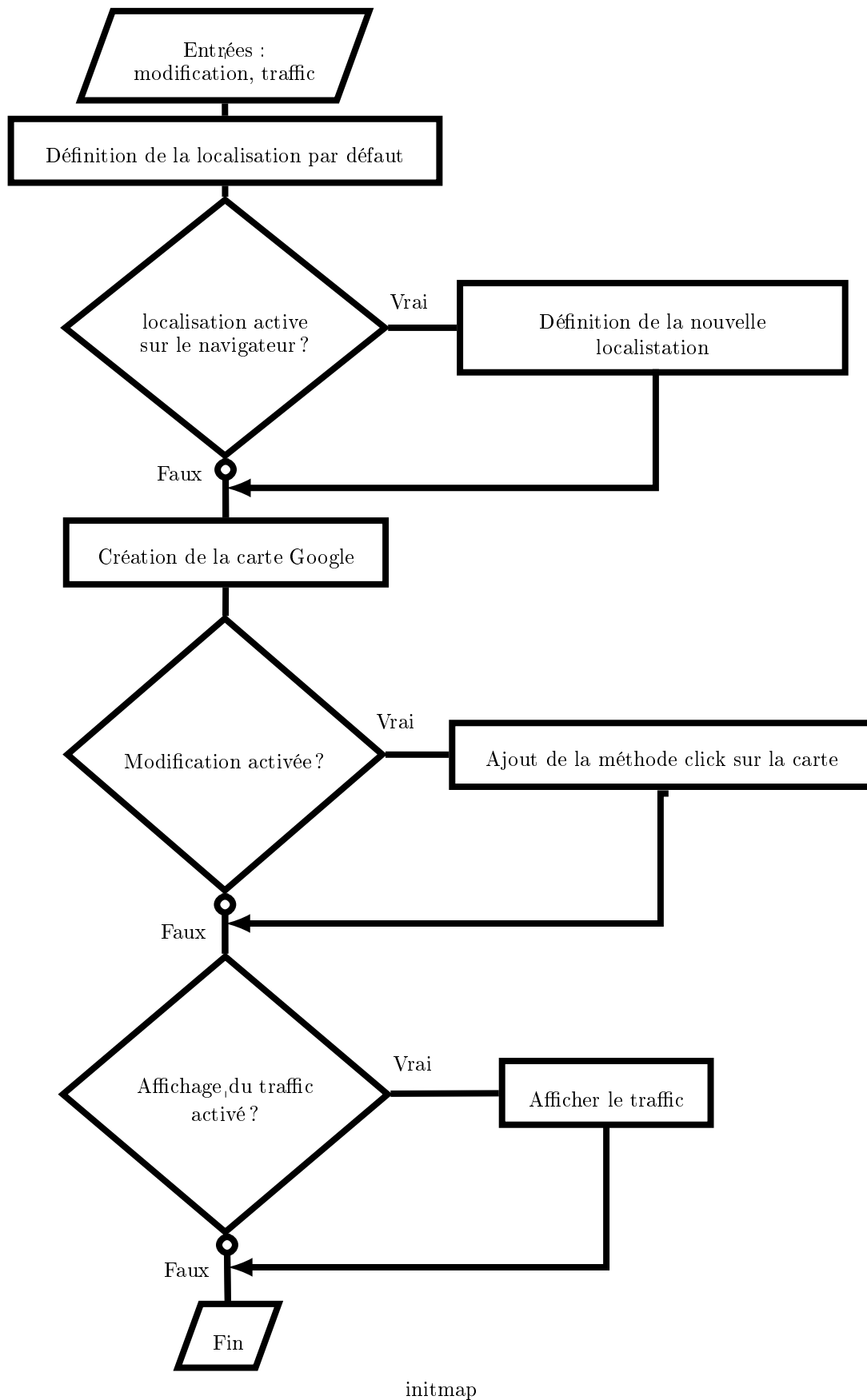
Lors de l'appel de l'initialisation il faut définir deux choses :

- Si le trajet peut être modifié
- Si le trafic doit être affiché

Par défaut la carte est centrée sur Genève. Si La géolocalisation est disponible et activée sur le navigateur, la carte se centre sur le lieu de l'utilisateur.

Si la variable de modification est vrai, un événement *click* sera ajouté à la carte.

De plus, si l'utilisateur choisi de visualiser le trafic en temps réel, un calque avec le trafic sera rejouté a *Google Maps*



### 5.10.2 ImportGPX

Prend en paramètre le fichier GPX à importer.

Fait un appel AJAX à la méthode d'import de fichier.

Ne retourne rien.

### 5.10.3 AskGoogle

Prend en paramètre le chemin de la route.

Fait un appel AJAX à l'API Google *snapToRoads*.

Récupère chaque valeurs données par l'API.

Retourne un tableau avec ces nouvelles valeurs.

### 5.10.4 LoadPoints

Prend en paramètre l'identifiant de la route.

Fait un appel AJAX à la fonction *GetRoutePoints*.

Récupère chaque valeur dans un tableau.

Retourne le tableau .

### 5.10.5 SnappPoints2Road

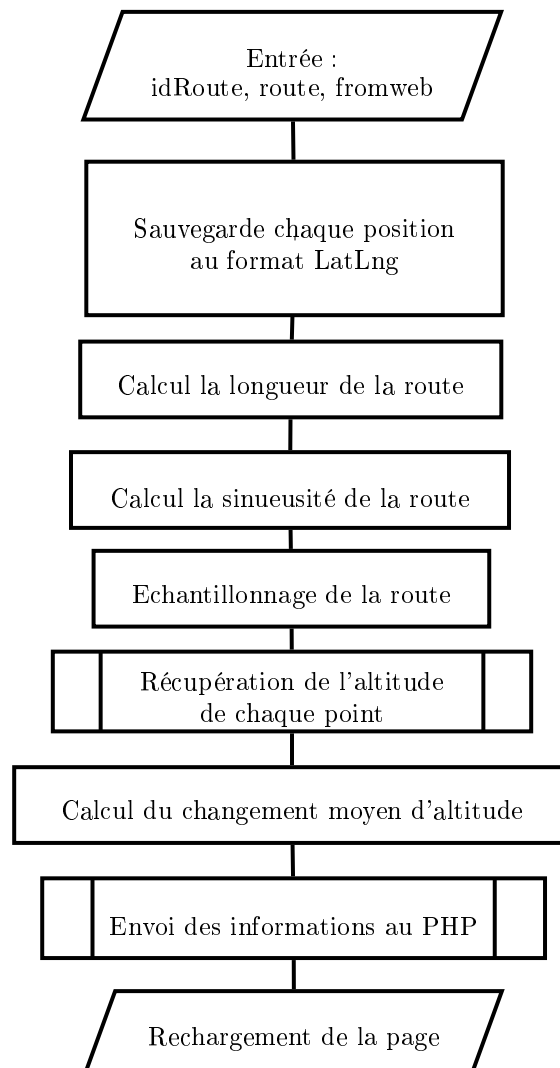
Prend en paramètre la route (sous fomrat d'un tableau de points).

Parcour le tableau et toute les 50 positions, appel la fonction *AskGoogle*

Retourne un tableau avec toutes le postions.

### 5.10.6 SaveNewLocation

Prend 3 paramètres, l'identifiant de la route, la route (sous fomrat d'un tableau de points), et un boolean pour savoir si la route vien depuis l'API google.



#### 5.10.7 CreateRoute

Prend 2 paramètres, le nom de la route et si elle contient des autoroutes.

Vérifie que le nom ne soit pas vide.

Fait un appel AJAX à la fonction *CreateRoute*.

Appel *SaveNewLocation* avec les valeurs reçus.

Ne retourne rien.

#### 5.10.8 RouteClick

Ne prend pas de paramètres. Ajoute un événement click sur tout les objets contenant la classe *route*.

Cet événement comprend :

Effacer les informations sur la carte.

Initialiser la carte sans modification.

Défini l'action du bouton *DeleteRoute* par la suppression de la route sélectionnée.

Enlève l'attribut *hidden* du bouton de modification du tracé.

Enlève l'attribut *hidden* des objects ayant pour classe *routeControl*.

Modifie l'attribut *name* du bouton *btnModif*.

Retire la classe *highlighted* de la route en surbrillance.

Ajoute la classe *highlighted* à la route actuelle.

Défini la variable *highlighted* avec cet objet.

Appel *ShowParcour*.

Appel *DisplayRouteInfo*.

ne retourne rien

### 5.10.9 searchIti

Prend en paramètre 2 tableaux de point GPS, le premier contient le lieu d'origine et le second contient la destination.

Crée une variable contenant la requête de recherche (origine, destination, mode de voyage, auto-routes?).

Défini la carte dans le service *Direction* de *Google*.

Cherche la route.

Ajoute chaque étape de la route dans la variable globale *route*.

Affiche la route.

Ne retourne rien.

### 5.10.10 Clear

Ne prend pas de paramètre.

Efface la tableau de marqueur.

Efface le tableau contenant la route.

Ne retourne rien.

### 5.10.11 ShowParcours

Prend en paramètre le tableau de points GPS du tracé, si le parcours peut être modifier et la couleur du parcours.

Si la couleur n'est pas définie, défini la couleur par défaut (*#ff1ece*).

Si le tracé peut être modifié, appel *CreateMarker* pour le marqueur de début et fin.

Ajout au tableau *route* les point du tracé.

Etand les "bornes" pour centrer la carte sur la totalité du tracé.

Appel la fonction *DisplayRoute*.

#### 5.10.12 DisplayRoute

Prend en paramètre la couleur du tracé.

Si aucun paramètre n'est saisi, définit la couleur avec la couleur par défaut (`#ff1eece`).

Affiche la *Polyline* du tracé sur la carte.

ne retourne rien.

#### 5.10.13 CreateMarker

Prend en paramètre, la localisation du marqueur, si la parcoure peut être modifiée, si le marqueur peut être supprimé, la couleur du marqueur, le texte du marqueur et sa position (ordre du tracé).

Définit l'image du marqueur selon la couleur et le texte choisis.

Crée le marqueur.

Si le marqueur peut être supprimé, ajoute les événements de suppression.

si le marqueur peut être modifié, ajoute les événements de modification.

Ajoute le marqueur au tableau de marqueurs.

Ne retourne rien.

#### 5.10.14 refreshValues

Cette fonction sert à rafraîchir la valeur d'un point GPS dans les zones de saisie.

#### 5.10.15 DownloadRoute

Ne prend pas de paramètres.

Vérifie qu'un itinéraire soit sélectionné.

Récupère le nom de l'itinéraire.

Fait un appel AJAX à la fonction PHP *Download* avec comme paramètres, le nom de la route et le chemin.

Fait appel à la page de téléchargement avec le nom du fichier.

ne retourne rien.

#### 5.10.16 EnableDisabledFilters

Prend en paramètre le checkbox.

Vérifie si le checkbox est checked ou non.

Ajoute ou retire l'attribut *Hidden*.

Si les filtres doivent être affichés, appelle *RefreshRouteWithFilters* sinon appelle *RefreshRouteWithoutFilters*.

#### 5.10.17 RefreshRouteWithFilters

Ne prend pas de paramètres.

Récupère la valeur de tout les filtres. (sinuosité, Pente et autoroute).

Fait un appel AJAX à la fonction PHP *FilterRoad* avec comme paramètres les valeurs récupérée précédemment.

Supprime l'affichage de toutes les routes.

Ajoute chaque route que l'AJAX nous a retourné.

Appel la fonction *RouteClick*

Ne retourne rien.

#### 5.10.18 RefreshRouteWithoutFilters

Ne prend pas de paramètres.

Fait un appel AJAX à la fonction PHP *GetRoutes*

Supprime l'affichage de toutes les routes.

Ajoutes les routes que l'AJAX nous a retourné.

Appel la fonction *RouteClick*.

Ne retourne rien.

#### 5.10.19 StartCreation

Ne prend pas de paramètres.

Appel la fonction *ClearRoute*.

Enlève la classe CSS *Hidden* à tout les objets qui ont pour classe *SaveRouteControls*.

#### 5.10.20 DisplayRouteInfo

Prend en paramètre l'identifiant de la route.

Fait un appel AJAX à la fonction PHP *GetRoadsInfos*.

Affiche toutes les informations retournée par l'appel AJAX dans les champs prévu à cet effet.

Ne retourne rien.

#### 5.10.21 DeleteRoute

Prend en paramètre l'identifiant de la route.

Fait un appel AJAX à la fonction PHP *DeleteRoute*.

Recharge la page.



### 5.10.22 ClearRoute

Ne prend pas de paramètre.

Appel la fonction *InitMap*

Vide tout les tableaux de données.

### 5.10.23 Document.Ready

Appel la fonction *initMap*.

Appel la fonction *Clear*.

Appel la fonction *RouteClick*.

Ajoute un événement click sur le bouton *btnmodif*.

Ajoute un événement click sur le bouton *RefreshRoute*.

Ajoute un événement lorsque la valeur du *slider* de sinueusité change.

Ajoute un événement lorsque la valeur du *slider* d'altitude change.

Ajoute un événement lorsque la case a cocher du filtre d'autoroute change.

Ajoute un événement click sur le bouton *StartCreation*.

Ajoute un événement click sur le bouton *ClearRoute*.

Ajoute un événement lorsque la case a cocher d'utilisation des autoroute change.

### 5.10.24 Document.bind

Ajoute une classe *loading* au body lorsque une requête AJAX démarre.

Retire la classe *loading* au body lorsque une requête AJAX se termine.

Ce système permet d'afficher une roue qui tourne pour faire patienter l'utilisateur.

## 5.11 fonction javascript pour l'inscription et la connexion

### 5.11.1 connexion

Prend un paramètre qui est un tableau contenant l'utilisateur et le mot de passe.

Fait un appel AJAX à la page PHP *Connexion*.

Vérifie la réponse AJAX et recharge la page ou affiche un message d'erreur.

ne retourne rien

### 5.11.2 LoadModalFromBrand

Prend en paramètre la marque.

Fait un appel AJAX à la fonction PHP *GetModel*

Affiche tout les model retourné par l'AJAX.

Ne retourne rien.

### 5.11.3 LoadYearFromModel

Prend 2 paramètres, la marque et le model.

Fait un appel AJAX à la fonction PHP *GetYear*

Affiche toutes les années retournée par l'AJAX.

Ne retourne rien.

### 5.11.4 UserNameExists

Prend en paramètre le nom d'utilisateur.

Crée une variable existe a faux.

Fait un appel AJAX à la fonction PHP *UserExists*.

Défini a vrai la variable si l'utilisateur existe.

Retourne la variable.

### 5.11.5 checkForEnabled

Ne prend pas de paramètre.

Vérifie pour chaque champs de saisie s'il y a la croix rouge. Si c'est la cas, desactive le bouton submit.

Ne retourne rien.

### 5.11.6 Document.Ready

Active les info-bulles.

Pour chaque champ de saisie, active le contrôle de saisie et mets a jour la croix rouge ou le vu vert.

## 5.12 Fonction ajax

### 5.12.1 GetModel

Prend en entrée la marque.

Sélectionne dans la base de données tous les model correspondant à la marque.

Retourne un tableau JSON contenant tous les models.

### 5.12.2 GetYear

Prend en entrée la marque et le model.

Sélectionne dans la base de données toutes les années d'un véhicule.

Retourne un tableau JSON contenant toutes les années.

#### **5.12.3 UserExists**

Prend en entrée le nom d'utilisateur.

Sélectionne dans la base de données l'utilisateur contenant ce nom d'utilisateur.

Si la base de données retourne quelque chose, retourne vrai sinon retourne faux

#### **5.12.4 GetMotorcycles**

Prend en entrée marque, model, année, consommation et indice de fatigue.

Sélectionne dans la base de données les motos correspondant aux critères saisis.

Retourne un tableau contenant toutes les informations des motos.

#### **5.12.5 GetRoutePoints**

Prend en entrée l'identifiant de la route.

Sélectionne dans la base de données tout les points GPS correspondant à l'identifiant de la route.

retourne un tableau contenant tous les points

#### **5.12.6 SaveNewRoute**

Prend en entrée l'identifiant de la route, la route, la sinuosité, l'altitude et la longueur.

Supprime les points enregistré pour cette route.

Ajoute la sinuosité.

Ajoute l'altitude.

Ajoute la longueur.

Vérifie le format d'entrée de la route, si c'est un string, le convertit en tableau.

Pour chaque points de la route, appelle la méthode d'ajout du point.

Ne retourne rien.

#### **5.12.7 AddMotorcycle**

Prend en entrée, la marque, le model, l'année, la consommation et l'indice de fatigue.

Vérifie que les champs ne soient pas nuls.

Ajoute la moto dans la base de données.

Retourne un tableau contenant soit un message d'erreur soit rien du tout.

#### **5.12.8 UpdateUserRole**

Prend en entrée, l'identifiant d'utilisateur et l'identifiant du futur rôle.

Vérifie que l'utilisateur connecté est administrateur.

Met à jours le rôle de l'utilisateur.

Retourne un tableau contenant soit une erreur soit le nouveau rôle de l'utilisateur.

#### **5.12.9 GetUserRole**

Prend en entrée, l'identifiant de l'utilisateur.

Appel la méthode GetUserRoleById.

Retourne le nom du rôle.

#### **5.12.10 downloadRoute**

Prend en entrée, le nom de la route et sont parcour.

Appel la fonction "Path2Gpx".

Retourne le nom du fichier.

#### **5.12.11 FilterRoad**

Prend en entrée tout les paramètres d'une route (sinueusité, pente, autoroute, durée).

Sélectionne dans la base de donnéesles routes qui correspondent aux critères saisi en entrée.

Retourne un tableau JSON avec toute les données.

#### **5.12.12 GetRoutesJSON**

Ne prend aucun paramètre.

Récupères toutes les routes en appelant la fonction *GetRoutes*.

Retourne les routes au format JSON.

#### **5.12.13 GetRoadsInfos**

Prend enentrée l'identifiant de la route.

Récupère dans la base de donnéesles informations de la route et, si l'utilisateur est connecté, récupère la consommation de sa moto.

Retourne un tableau JSON avec les informations.

#### **5.12.14 CreateNewRoute**

Prend en entrée le nom de la nouvelle route ainsi que si elle contient des autoroutes.

Appel la méthode CreateRoute.

Retourne l'identifiant de la nouvelle route au format JSON.

**5.12.15 DeleteRoute**

Prend en paramètre l'identifiant de la route.

Supprime dans la base de données les lieux comportant l'identifiant de la route ainsi que la route.

Retourne "Success" au format JSON

## 5.13 Base de données

La moteur de base de données utilisé est *l'InnoDB*, afin de garder les valeur en *UTF-8*. Le système de gestion de la base de données est *MySQL*.

### 5.13.1 Modèle conceptuel

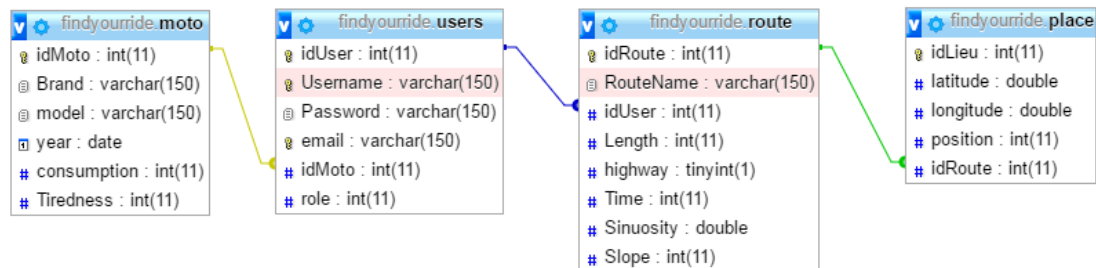


FIGURE 7 – Modèle conceptuel de données

### 5.13.2 Structure de la base de données

#### 5.13.2.1 Table *moto*

Cette table regroupe toutes les motos du site.

Colone	Type	Description
<b>idMoto</b>	int	Identifiant de la moto
Brand	Varchar(150)	Marque de la moto
model	Varchar(150)	Modèle de la moto
year	date	Date de sortie de la moto
consumption	int	Consommation de la moto
Tiredness	int	Indice de fatigue de la moto

#### 5.13.2.2 Table *users*

Cette table regroupe tous les utilisateurs du site.

Colone	Type	Description
<b>iduser</b>	int	Identifiant de l'utilisateur
Username	Varchar(150)	Nom de l'utilisateur
Password	Varchar(150)	Mot de passe de l'utilisateur (sha1)
email	Varchar(150)	E-mail de l'utilisateur
#idMoto	int	Identifiant de la moto
role	int	Role de l'utilisateur

#### 5.13.2.3 Table *route*

Cette table regroupe toutes les routes du site.

Colone	Type	Description
<b>idRoute</b>	int	Identifiant de la route
RouteName	Varchar(150)	Nom de la route
#idUser	int	Identifiant de l'utilisateur créateur de la route
Length	int	Longueur de la route (en mètres)
Highway	Tinyint	La route contient des autoroutes ?
Time	int	Durée de la route
Sinuosity	double	Sinueusité de la route
Slope	double	Pente moyenne de la route

#### 5.13.2.4 Table *Place*

Cette table regroupe tous les points géographique du site.

Colone	Type	Description
<b>idLieu</b>	int	Identifiant du lieu
latitude	double	Latitude de l'endroit
longitude	double	Longitude de l'endroit
position	int	position dans le tracé
#idRoute	int	Identifiant de la route correspondante au lieu

## 6 Estimation de l'apport personnel

Fichiers	Détails	pourcentage
*.php	J'ai réalisé la totalité des pages PHP	100%
*.css	J'ai réalisé une partie du code CSS nécessaire pour le positionnement de certain objet sinon le reste est fait par <i>bootstrap</i>	5%
*.js	J'ai réaliser la totalité des fontion de gestion pour google map ormis les fonctions d'affichage des lignes modifiable. Le reste des fichier sont ceux de <i>bootstrap</i> et je ne les ai pas modifié	80%



## 7 Problèmes rencontré

### 7.1 Google Api

Durant ce projet, j'ai été confronté à l'utilisation des API Google.

Ces API sont bien documentée et il est facile de comprendre comment les utiliser.

Cependant Google a établi des quotas assez stricte ce qui fait qu'on se retrouve vite limité.

Il m'est arrivé a plusieurs reprise de ne plus pouvoir faire de requête car j'avais atteint le nombre maximal de requête.

Pour remedier à ce problème, j'ai créé plusieurs compte developpeur Google et lorsque l'un de mes compte arrivais à la limite des quotas je changeais de clef d'API.

Cette méthode n'est néanmoins pas optimal et il n'est pas possible de publier un site web dans ces conditions.

La solution est de payer un abonnement auprès de Google afin d'obtenir plus de requête journalière.

### 7.2 Asynchrone

Les fonctions de rappel javascript fonctionnent toutes de façon asynchrone.

J'ignorais cet aspect au début du travail, ce qui fait qu'il m'a fallut un peu de temps pour comprendre pourquoi certaine variables changeais de valeur "toute seule" ou avait des valeurs différente en mode deboguage.

Pour l'AJAX, lorsque j'avais besoin qu'une fonction de rappel ne soit pas asynchrone, j'utilisait le champ *async* que je définissait a *false*.

## 8 Conclusion

### 8.1 Bilan personnel

Durant ce projet de 9 semaines, j'ai pu découvrir les outils de développement Google. Ceux-ci sont très complet et relativement simple d'utilisation grâce à leurs bonne documentation.

Je regrette de ne pas avoir fait un bon planning dès le début car je me suis perdu arrivé à environ la moitié du projet. Une bonne planification m'aurait permis de travailler plus efficacement en suivant un fil conducteur.

L'objectif de mon travail qui était *"avoir un site web permettent le partage entre les passionnés de la moto"* est atteint.

### 8.2 Améliorations

Au fur et à mesure du travail des idées me sont apparue pour l'amélioration du site :

- Permettre aux utilisateur de noter un itinéraire.
- Fournir à l'utilisateur un lien unique pour partager un itinéraire avec des personnes externe au site.
- Permettre à l'utilisateur d'avoir plusieurs moto et de les gérer
- Ajouter la "fatigue" sur les itinéraires

## Table des figures

1	Schéma du site . . . . .	8
2	Page inscription . . . . .	10
3	Page inscription . . . . .	10
4	Page Detail Trajet . . . . .	11
5	Modal connexion . . . . .	11
6	Page vos trajet . . . . .	12
7	Modèle conceptuel de données . . . . .	37