

*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

8 juin 2017

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Résumé . . . . .	4
1.2	Abstract . . . . .	4
<b>2</b>	<b>Cahier des charges</b>	<b>5</b>
<b>3</b>	<b>Analyse de l'existant</b>	<b>6</b>
3.1	Garmin BaseCamp . . . . .	6
3.2	<a href="http://www.calculitineraires.fr/">http://www.calculitineraires.fr/</a> . . . . .	6
3.3	<a href="http://www.bestbikingroads.com">http://www.bestbikingroads.com</a> . . . . .	6
3.4	Conclusion . . . . .	6
<b>4</b>	<b>Analyse fonctionnelle</b>	<b>7</b>
4.1	Généralités . . . . .	7
4.2	Description des fonctionnalités globales . . . . .	8
4.2.1	Connexion . . . . .	8
4.2.2	Inscription . . . . .	8
4.2.3	Création de trajet . . . . .	8
4.2.4	Exportation . . . . .	8
4.2.5	Suppression d'un trajet . . . . .	8
4.2.6	Visualiser le trajet . . . . .	8
4.3	Description de l'interface . . . . .	9
4.3.1	Inscription . . . . .	9
4.3.2	Map . . . . .	9
4.3.3	Detail trajet . . . . .	10
4.3.4	Connexion . . . . .	10
4.3.5	Vos trajets . . . . .	11
4.4	Description des éléments de sécurité . . . . .	11
4.4.1	Fichier .htaccess . . . . .	11
4.4.2	Utilisateur de la base de données . . . . .	11
4.4.3	Requêtes . . . . .	11
<b>5</b>	<b>Analyse organique</b>	<b>12</b>
5.1	Résumé . . . . .	12

5.2	Google API . . . . .	12
5.2.1	Google Maps Javascript . . . . .	12
5.2.2	Google Direction . . . . .	12
5.2.3	Google Elevation . . . . .	12
5.2.4	Google Roads . . . . .	12
5.3	Fichier GPX . . . . .	13
5.4	Fonction PHP pour la base de données . . . . .	13
5.4.1	connexion à la base de données . . . . .	13
5.4.2	Requêtes préparées . . . . .	13
5.4.3	GetMotorcycleBrand . . . . .	14
5.4.4	GetMotorcycleModel . . . . .	14
5.4.5	GetAllMotorcycleYear . . . . .	14
5.4.6	GetAllMotorcycleConsumption . . . . .	14
5.4.7	GetAllMotorcycleTiredness . . . . .	14
5.4.8	GetAllMotorcycles . . . . .	14
5.4.9	GetRoutes . . . . .	14
5.4.10	CreateRoute . . . . .	15
5.4.11	deletePlaces . . . . .	15
5.4.12	addPlaceToRoute . . . . .	15
5.4.13	deleteMotorcycle . . . . .	15
5.4.14	getUserNMotorcycle . . . . .	15
5.4.15	getUserRoleById . . . . .	15
5.4.16	AddSinuosity . . . . .	15
5.4.17	AddElevation . . . . .	16
5.4.18	AddLength . . . . .	16
5.4.19	GetmostSinuousRoad . . . . .	16
5.4.20	GetLessSinuousRoad . . . . .	16
5.4.21	GetMostSteepestRoad . . . . .	16
5.4.22	GetLessSteepestRoad . . . . .	16
5.5	Inscription sur le site . . . . .	17
5.6	Connexion au site . . . . .	18
5.7	Upload d'un itinéraire . . . . .	20
5.8	Téléchargement d'un itinéraire . . . . .	21
5.9	Fonction javascript de gestion de la carte . . . . .	22
5.9.1	Initialisation de la carte . . . . .	22

---

5.9.2	ImportGPX . . . . .	25
5.9.3	AskGoogle . . . . .	25
5.9.4	LoadPoints . . . . .	25
5.9.5	SnappPoints2Road . . . . .	25
5.9.6	SaveNewLocation . . . . .	25
5.9.7	CreateRoute . . . . .	26
5.9.8	RouteClick . . . . .	26
5.10	Fonction ajax . . . . .	27
5.10.1	GetModel . . . . .	27
5.10.2	GetYear . . . . .	27
5.10.3	UserExists . . . . .	27
5.10.4	GetMotorcycles . . . . .	27
5.10.5	GetRoutePoints . . . . .	27
5.10.6	SaveNewRoute . . . . .	28
5.10.7	AddMotorcycle . . . . .	28
5.10.8	UpdateUserRole . . . . .	28
5.10.9	GetUserRole . . . . .	28
5.10.10	downloadRoute . . . . .	28
5.10.11	FilterRoad . . . . .	29
5.10.12	GetRoutesJSON . . . . .	29
5.10.13	GetRoadsInfos . . . . .	29
5.10.14	CreateNewRoute . . . . .	29
5.10.15	DeleteRoute . . . . .	29
5.11	Base de données . . . . .	30
5.11.1	Modèle conceptuel . . . . .	30
5.11.2	Structure de la base de données . . . . .	30
<b>6</b>	<b>Problèmes rencontré</b>	<b>32</b>
6.1	Google Api . . . . .	32
6.2	Async . . . . .	32
<b>7</b>	<b>Conclusion</b>	<b>32</b>

# 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**

TODO inclure le CDC

### 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 possibilité à 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 recherche 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 recherche, 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 même temps sur la carte</li><li>• pas de modification possible</li></ul>

#### 3.4 Conclusion

Il existe différents sites de partage d'itinéraire, cependant ils ont tous des fonctionnalités assez similaires. Mon site se distingue des autres de par son interface intuitive pour l'utilisateur et certaines fonctionnalités trouvées 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 leur 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.

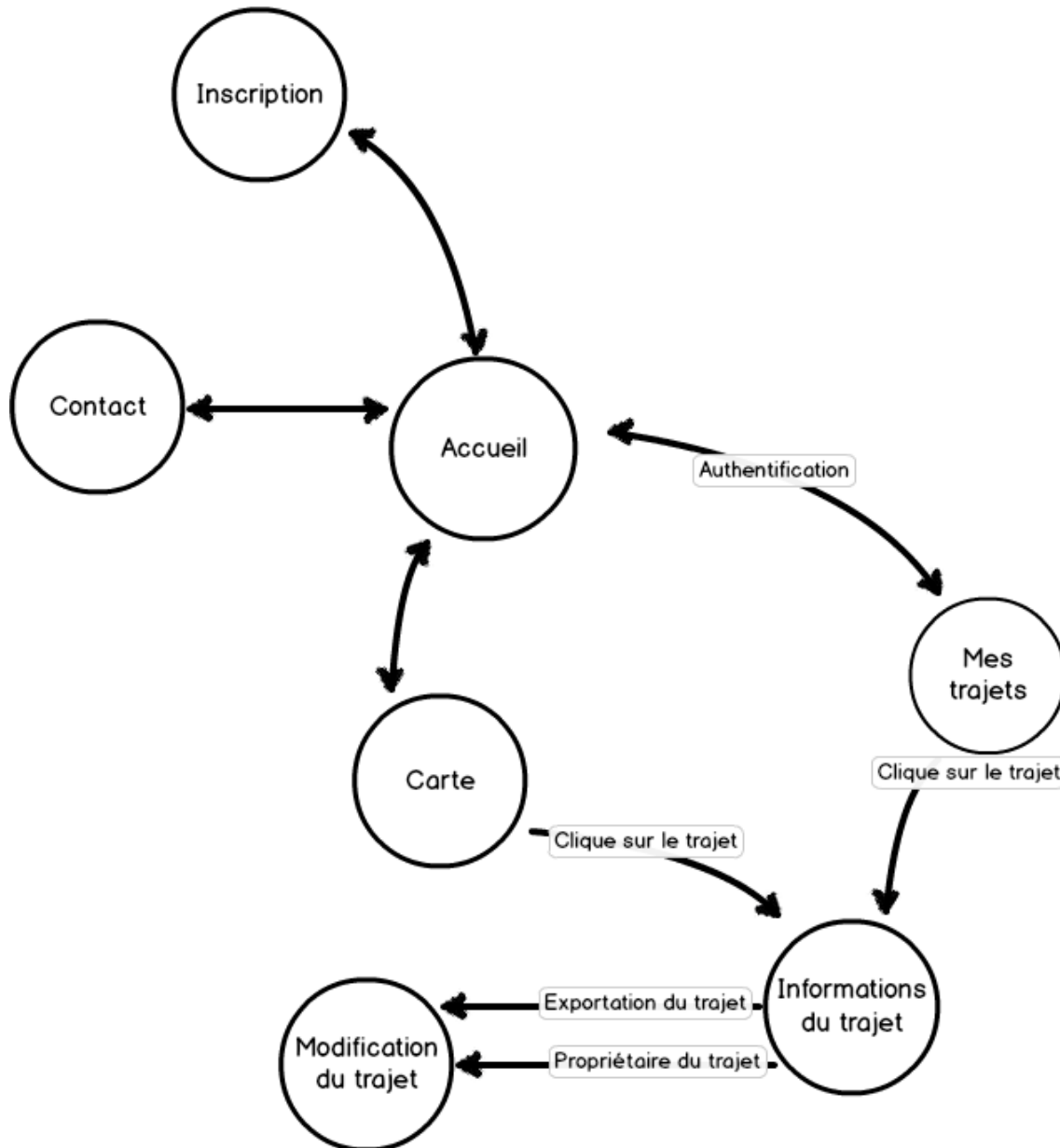


FIGURE 1 – Schéma du site



## 4.2 Description des fonctionnalités globales

### 4.2.1 Connexion

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.

## 4.3 Description de l'interface

### 4.3.1 Inscription

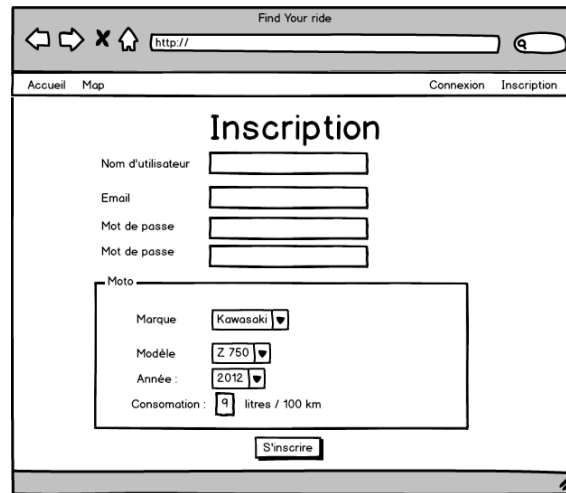


FIGURE 2 – Page inscription

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. Le champ *Moto* est facultatif mais conseillé.

### 4.3.2 Map

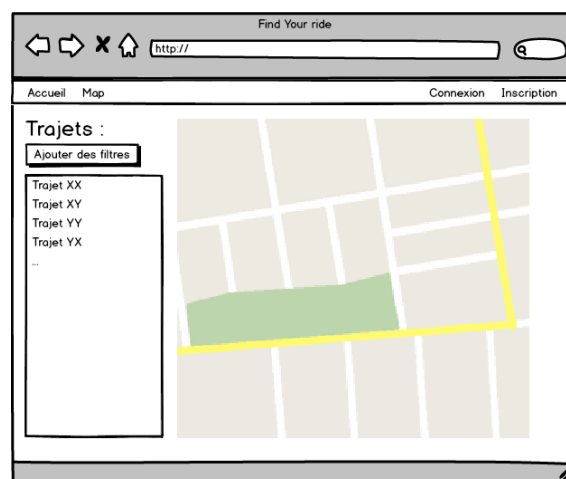


FIGURE 3 – Page inscription

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

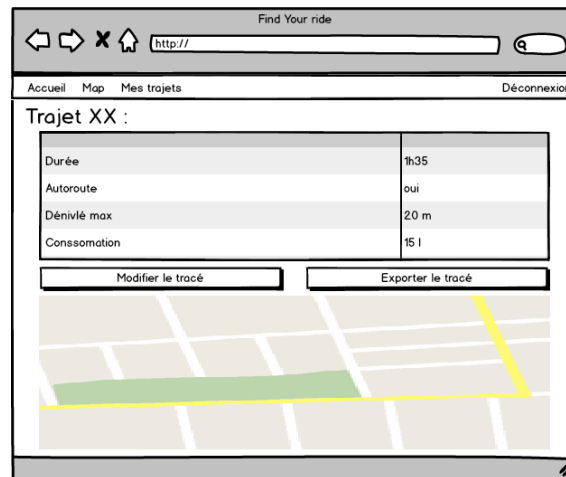


FIGURE 4 – Page Detail Trajet

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

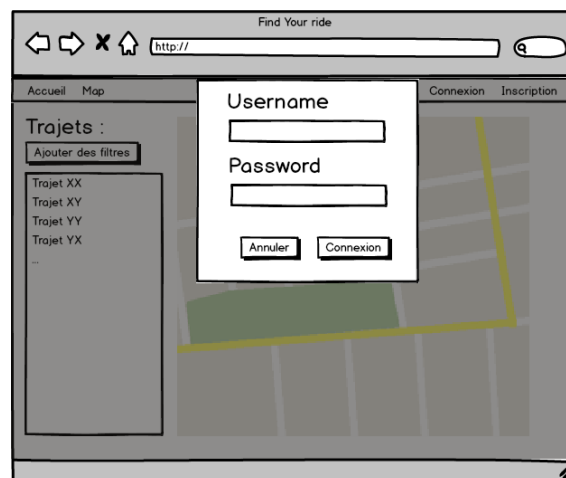


FIGURE 5 – Modal connexion

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

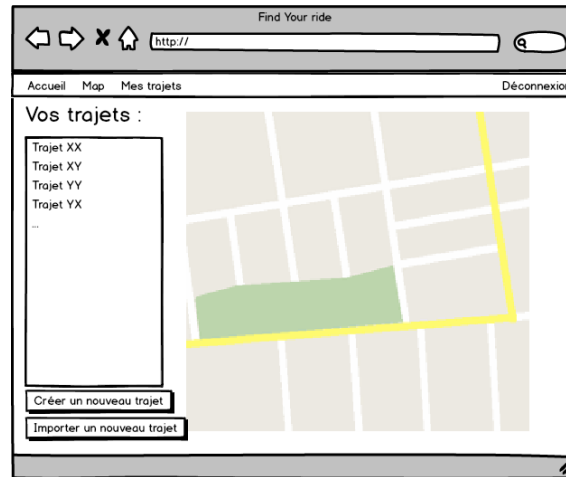


FIGURE 6 – Page vos trajet

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 expliquer le fonctionnement de l'application.

### 5.2 Google API

Les API<sup>2</sup> Google sont une suite d'outils développé et fourni 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

#### 5.2.1 Google Maps Javascript

Cette API permet d'intégrer une *Google Map* 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.

#### 5.2.2 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 différent paramètres.

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

#### 5.2.3 Google Elevation

TODO

#### 5.2.4 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 point GPS de notre route et il nous retourne un ensemble de point qui suivent la route.

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

---

<sup>2</sup>. *Application Programming Interface* (Interface de Programation Applicative) [https://fr.wikipedia.org/wiki/Interface\\_de\\_programmation](https://fr.wikipedia.org/wiki/Interface_de_programmation)

## 5.3 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 </trkpt>
16 </trk>
```

## 5.4 Fonction PHP pour la base de données

### 5.4.1 connexion à la base de données

La connexion à la base de données se fait à l'aide de PDO<sup>3</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>4</sup>.

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

### 5.4.2 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.

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

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

#### **5.4.3 GetMotorcycleBrand**

Ne prend pas de paramètre d'entrée.

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

Retourne un tableau contenant les marques.

#### **5.4.4 GetMotorcycleModel**

Ne prend pas de paramètre d'entrée.

Récupère dans la base de données tout les models de moto.

Retourne un tableau contenant les models trié par ordre alphabétique.

#### **5.4.5 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.

#### **5.4.6 GetAllMotorcycleConsumption**

Ne prend pas de paramètre d'entrée.

Récupère dans la base de données toutes les consommations.

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

#### **5.4.7 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.

#### **5.4.8 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.

#### **5.4.9 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.

#### 5.4.10 CreateRoute

Prend 3 données en entrée, 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.

#### 5.4.11 deletePlaces

Prend en entrée l'identifiant de la route.

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

Ne retourne rien

#### 5.4.12 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

#### 5.4.13 deleteMotorcycle

Prend en entrée l'identifiant de la moto

Supprime la moto

Ne retourne rien.

#### 5.4.14 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.

#### 5.4.15 getUserRoleById

Prend en entrée l'identifiant de l'utilisateur.

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

Retourne le rôle.

#### 5.4.16 AddSinuosity

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

Ajoute la sinuosité dans la base de données



Ne retourne rien.

#### **5.4.17 AddElevation**

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

Ajoute l'altitude dans la base de données

Ne retourne rien.

#### **5.4.18 AddLength**

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

Ajoute la longueur dans la base de données

Ne retourne rien

#### **5.4.19 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.

#### **5.4.20 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.

#### **5.4.21 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".

#### **5.4.22 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".

## 5.5 Inscription sur le site

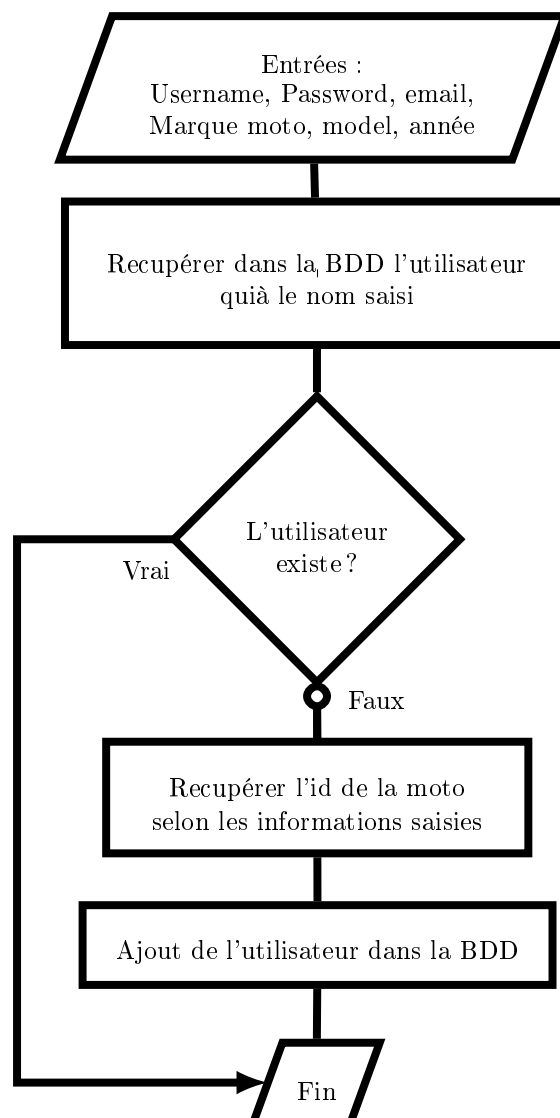
Afin de pouvoir bénéficier de toutes les fonctionnalités du site, l'utilisateur doit se connecter. Si celui-ci n'a pas de compte, il a la possibilité d'en créer un.

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.



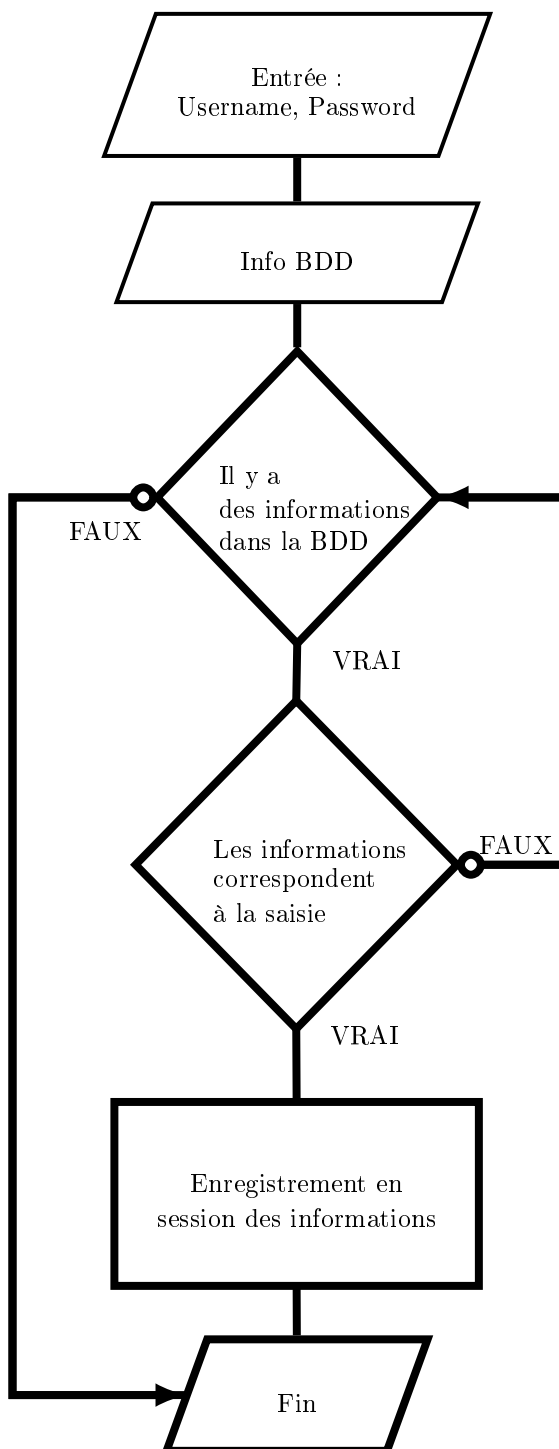
## 5.6 Connexion au site

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

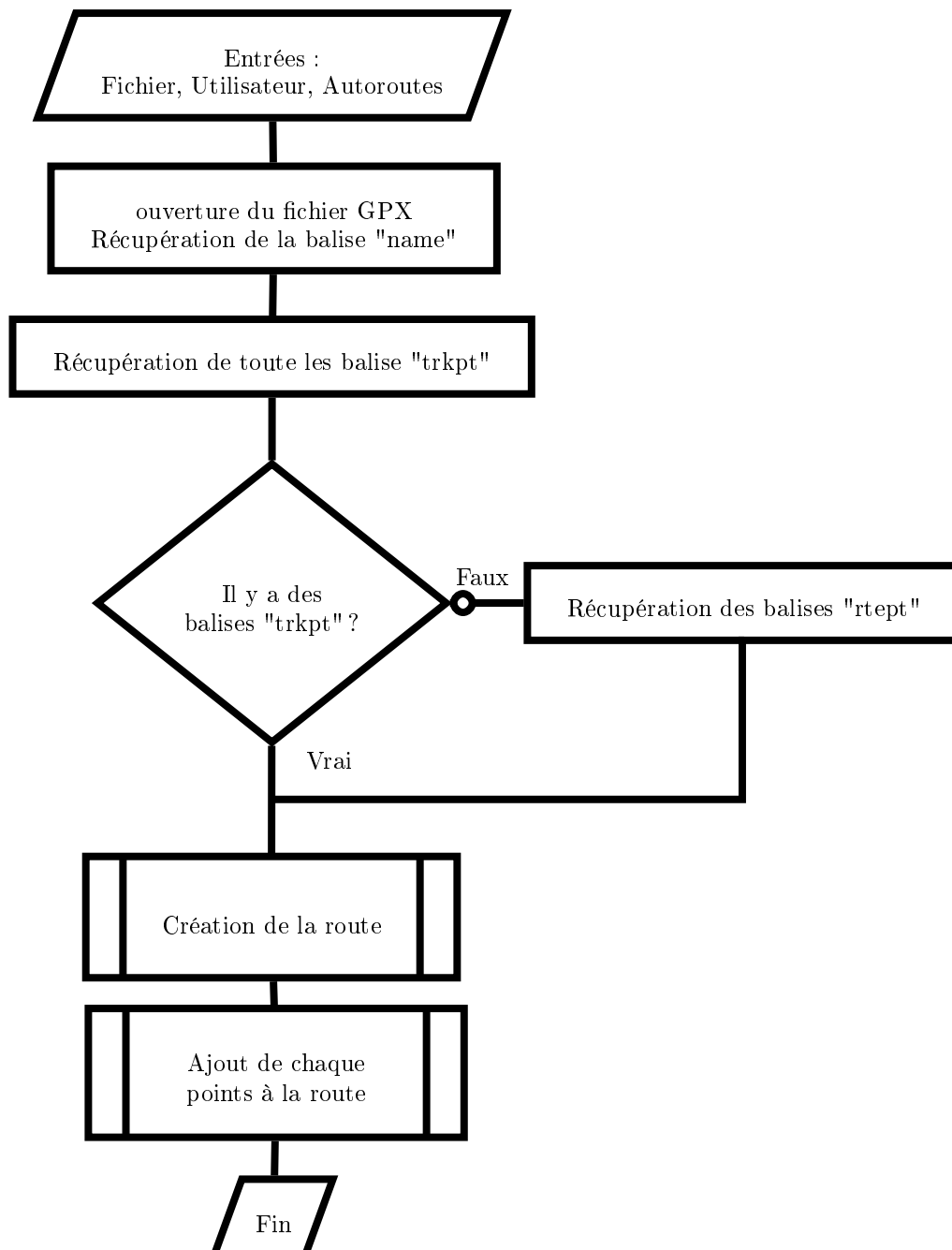
Une fois ces informations récupérée, on tous les parcourir pour être sûr que le nom d'utilisateur et le mot de passe fourni correspondent bien à un utilisateur existant. Si c'est le cas, on enregistre l'id, le role et le nom d'utilisateur dans des *Sessions* et on retourne *true* pour dire que les informations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



## 5.7 Upload d'un itinéraire

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

Premièrement, 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 à google les points et récupère la position sur la route (API Google Roads) et enregistre les nouvelles coordonnées.



## 5.8 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*.

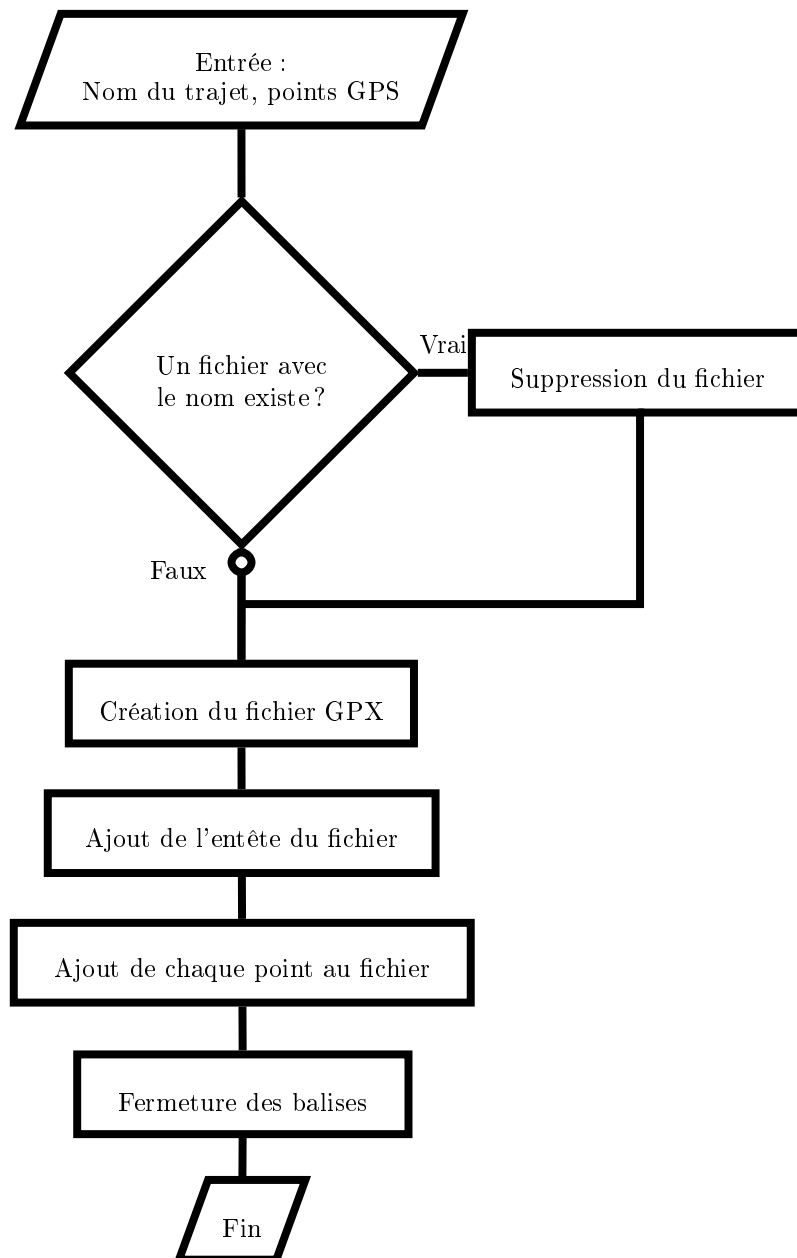
Le traitement se fait en PHP.

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.



## 5.9 Fonction javascript de gestion de la carte

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

### 5.9.1 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 :

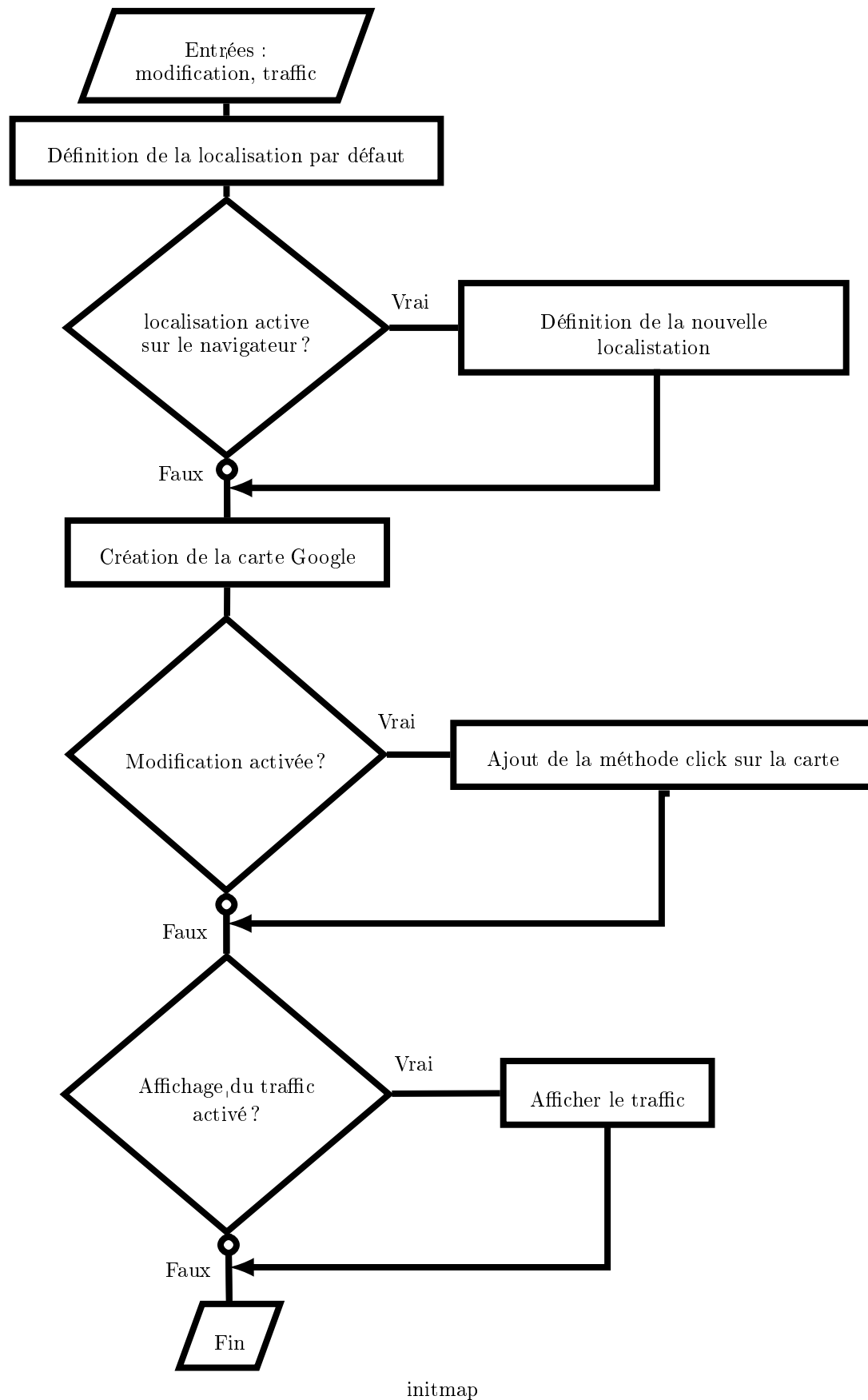
- Si le trajet peut être modifié
- Si le trafic doit être affiché

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 rajouté a *Google Maps*





initmap

### 5.9.2 ImportGPX

Prend en paramètre le fichier GPX à importer.

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

Ne retourne rien.

### 5.9.3 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.

### 5.9.4 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 .

### 5.9.5 SnappPoints2Road

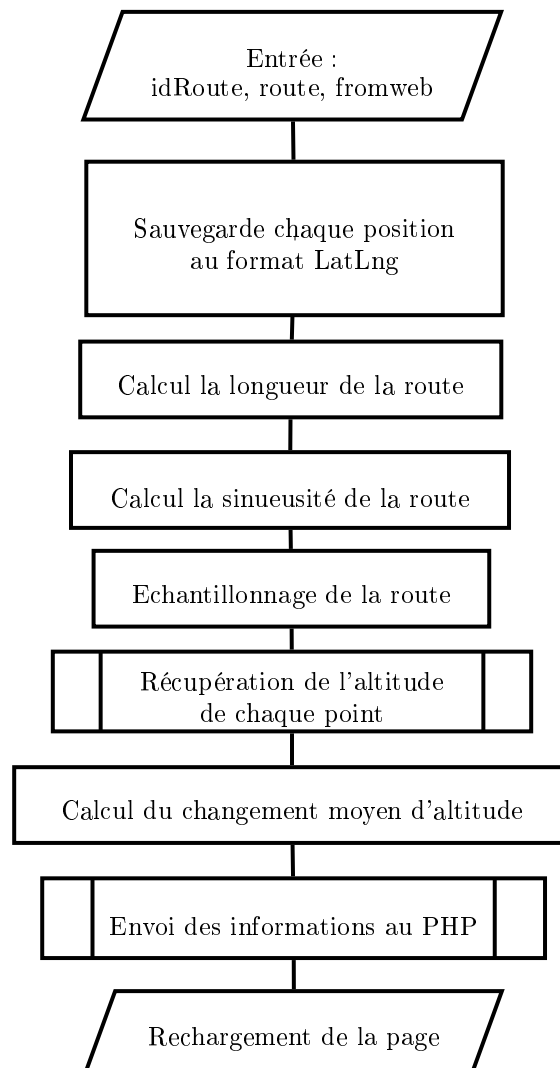
Prend en paramètre la route (sous fomrat d'un tableau de points).

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

Retourne un tableau avec toutes le postions.

### 5.9.6 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.



### 5.9.7 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.

### 5.9.8 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.

Définit 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 objets 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.

Définit la variable *highlighted* avec cet objet.

Appel *ShowParcour*.

Appel *DisplayRouteInfo*.

ne retourne rien

## 5.10 Fonction ajax

### 5.10.1 GetModel

Prend en entrée la marque.

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

Retourne un tableau JSON contenant tous les modèles.

### 5.10.2 GetYear

Prend en entrée la marque et le modèle.

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

Retourne un tableau JSON contenant toutes les années.

### 5.10.3 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

### 5.10.4 GetMotorcycles

Prend en entrée marque, modèle, 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.

### 5.10.5 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

#### **5.10.6 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.

#### **5.10.7 AddMotorcycle**

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

Vérifie que les champs ne soient pas nul.

Ajoute la moto dans la base de données.

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

#### **5.10.8 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.

#### **5.10.9 GetUserRole**

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

Appelle la méthode GetUserRoleById.

Retourne le nom du rôle.

#### **5.10.10 downloadRoute**

Prend en entrée, le nom de la route et son id.

Appelle la fonction "Path2Gpx".

Retourne le nom du fichier.

**5.10.11 FilterRoad**

Prend en entrée tout les paramètres d'une route (sinuosité, 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.

**5.10.12 GetRoutesJSON**

Ne prend aucun paramètre.

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

Retourne les routes au format JSON.

**5.10.13 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.

**5.10.14 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.

**5.10.15 DeleteRoute**

Prend en paramètre l'identifiant de la route.

Supprime dans la base de donnéesles lieu comportant l'identifiant de la route ainsi que la route.

Retourne "Success" au format JSON

## 5.11 Base de données

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

### 5.11.1 Modèle conceptuel

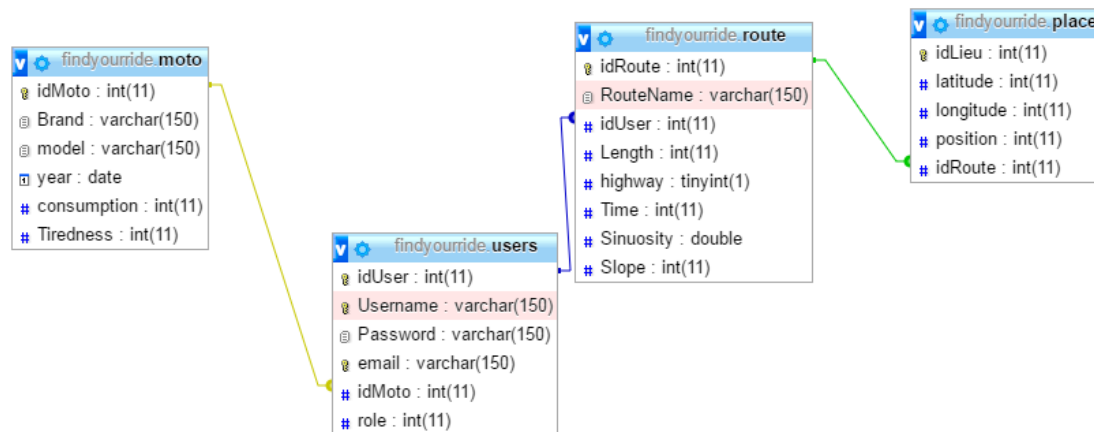


FIGURE 7 – Modèle conceptuel de données

### 5.11.2 Structure de la base de données

#### 5.11.2.1 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

#### 5.11.2.2 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	Rôle de l'utilisateur

### 5.11.2.3 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

### 5.11.2.4 Table *Place*

Cette table regroupe tous les points géographique du site.

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



## 6 Problèmes rencontré

### 6.1 Google Api

### 6.2 Async

## 7 Conclusion

## Table des figures

1	Schéma du site . . . . .	7
2	Page inscription . . . . .	9
3	Page inscription . . . . .	9
4	Page Detail Trajet . . . . .	10
5	Modal connexion . . . . .	10
6	Page vos trajet . . . . .	11
7	Modèle conceptuel de données . . . . .	30