

# Runscape Web App



*Philibert, Alexandre  
Rue du temple 8  
1148, Cuarnens*

## Table des matières

1	Analyse préliminaire .....	3
1.1	Introduction .....	3
1.2	Objectifs.....	3
1.3	Planification initiale .....	6
2	Analyse.....	6
2.1	Création d'activités .....	10
2.2	Modèle conceptuel des données .....	11
2.3	Stratégie de test.....	11
2.4	Budget initial .....	11
3	Conception .....	12
3.1	Ressources de l'API .....	12
3.2	Endpoints.....	12
3.3	Gestion lieu et pays .....	13
3.4	Gestion des fichiers GPX.....	14
3.5	Intégration du middleware Formidable.....	14
3.6	Modèle logique des données .....	15
3.7	Cas d'utilisations .....	19
4	Réalisation.....	20
4.1	Description des tests effectués .....	21
5	Annexes.....	22
5.1	Sources .....	22

## 1 Analyse préliminaire

### 1.1 Introduction

Le projet consiste à enrichir fonctionnellement un prototype existant par la création d'une interface graphique permettant de dialoguer avec un back-end existant. Ce projet fait suite à la création d'une API permettant la création, lecture, modification et suppression d'utilisateurs, de types d'activités et d'activités sportives. L'API intègre également un mécanisme de création de token d'authentification. J'ai choisi d'entreprendre ce projet car je souhaiterais ajouter un composant graphique au travail que j'ai déjà pu effectuer.

Les fonctionnalités existantes sur l'application étant reprises pour la réalisation de ce projet sont :

- Standardisation du retour d'erreur de l'API.
- Authentification par session et json web token.
- Importation d'une activité.
- Profil administrateur.
- Un effort particulier a été produit pour obtenir une architecture cohérente et faciliter les futures évolutions.

### 1.2 Objectifs

#### 1.2.1 **Création d'activité - sans gpx (Story 001)**

En tant que sportif (membre), je veux créer des activités, afin de suivre ma progression.

#### **Test d'acceptations**

##### Contexte

En tant que membre (sportif), je navigue jusqu'à la page me permettant de créer une nouvelle activité. Je ne dispose pas de gpx.

##### Événement

Je crée l'activité en saisissant les attributs suivants :

- L'heure de départ
- Le type d'activités (vélo, natation, course à pied, marche)
- La durée de l'effort
- Distance parcourue
- Lieu et pays de réalisation de l'activité
- Les dénivelés positifs et négatifs
- La vitesse moyenne

## Critères de réussite

Le système enregistre chacune des dimensions mentionnées ci-dessus et affiche l'activité.

### **1.2.2 Création d'activité - avec gpx (Story 002)**

En tant que sportif (membre), je veux créer des activités, afin de pouvoir suivre ma progression.

#### **Test d'acceptations**

##### Contexte

Je navigue jusqu'à la page me permettant de créer une nouvelle activité. Je dispose d'un gpx.

##### Événement

Je crée l'activité en important le gpx et saisissant manuellement les données qui ne sont pas présente ou interprétable depuis le gpx.

- Tous les attributs présents dans la story 001 (sans gpx) doivent être traités.
- Le parcours.

##### Critères de réussite

Le système enregistre chacune des dimensions mentionnées ci-dessus et affiche l'activité.

Note : la représentation graphique du parcours fait partie d'une story séparée.

## 1.2.3 Interprétation du parcours (Story 003)

En tant que sportif, je veux pouvoir observer mon entraînement sur une carte, afin d'analyser plus précisément mes performances.

### Test d'acceptations

#### Contexte

Je dispose d'au moins d'une activité qui a été créer à l'aide d'un gpx. Je navigue jusqu'à la page me permettant d'afficher l'activité de manière graphique.

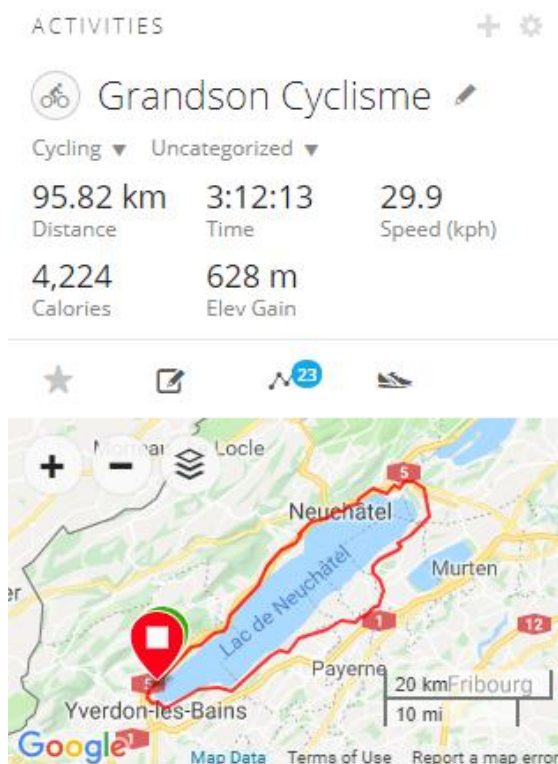
#### Evénement

Je demande à afficher mon parcours de manière graphique.

#### Critères de réussite

Le système affiche le parcours. Une carte (topologique, satellite) est présente avec une représentation du parcours.

Soit en déplaçant le curseur sur le parcours, soit un tableau doivent me permettre de connaître le "pace" (moyenne au kilomètre) pour chaque kilomètre.



Exemple d'IHM : Source garmin connect

## 1.3 Planification initiale

	i	Mode Tâche ▾	Nom de la tâche ▾	Durée ▾	Début ▾	Fin ▾	Prédécesseurs ▾
1		★	Début TPI	0 jour	Mar 07.05.19	Mar 07.05.19	
2		🔗	Planification initiale	1,5 hr	Mar 07.05.19	Mar 07.05.19	1
3		🔗	Mise en place de l'environnement de travail	4 hr	Mar 07.05.19	Mar 07.05.19	2
4		🔗	Entretien avec Monsieur Glassey	0,5 hr	Mar 07.05.19	Mar 07.05.19	3
5		🔗	Analyse et conception des changements à apporter sur la création d'activités d'un point de vue back-end	4 hr	Mar 07.05.19	Mer 08.05.19	4
6		🔗	Adaptation du endpoint de création d'activité pour respecter les spécifications des stories 001 et 002	6 hr	Mer 08.05.19	Jeu 09.05.19	5
7		🔗	Analyse et conception d'une solution permettant l'envoi de fichiers plus volumineux à l'API	5 hr	Jeu 09.05.19	Ven 10.05.19	6
8		🔗	Implémentation de la solution sélectionnée pour l'envoi de fichiers volumineux sur l'API	6 hr	Ven 10.05.19	Mar 14.05.19	7
9		🔗	Entretien avec Monsieur Glassey	1 hr	Mar 14.05.19	Mar 14.05.19	8
10		🔗	Analyse et conception de l'interface web de création d'activité spotives	7 hr	Mar 14.05.19	Jeu 16.05.19	9
11		🔗	Implémentation de l'interface web de création d'activité sportives	8 hr	Jeu 16.05.19	Ven 17.05.19	10
12		🔗	Analyse et conception d'une interface permettant la sélection d'une activité enregistrée dans l'application	3 hr	Ven 17.05.19	Ven 17.05.19	11
13		🔗	Implémentation de l'interface permettant la sélection d'une activité	8 hr	Ven 17.05.19	Mar 21.05.19	12
14		🔗	Entretien avec Monsieur Glassey	1 hr	Mer 22.05.19	Mer 22.05.19	13
15		🔗	Analyse et conception de l'interprétation du parcours (Story 003)	6 hr	Mer 22.05.19	Jeu 23.05.19	14
16		🔗	Création de l'interface web d'affichage des activités en excluant la carte	5 hr	Jeu 23.05.19	Ven 24.05.19	15
17		🔗	Intégration de la carte à l'interface graphique	12 hr	Ven 24.05.19	Mer 29.05.19	16
18		🔗	Entretien avec Monsieur Glassey	1 hr	Mer 29.05.19	Mer 29.05.19	17
19		🔗	Analyse et conception de l'affichage de la moyenne de vitesse au kilomètre	3 hr	Mer 29.05.19	Mar 04.06.19	18
20		🔗	Implémentation de la solution choisie pour l'affichage de la moyenne de vitesse au kilomètre	6 hr	Mar 04.06.19	Mar 04.06.19	19
21		🔗	Réalisation d'un tutoriel vidéo	2 hr	Mar 04.06.19	Mer 05.06.19	20
22		🔗	Réalisation du résumé du rapport de travail	2 hr	Mer 05.06.19	Mer 05.06.19	21
23		🔗	Réalisation du livrable final	3 hr	Mer 05.06.19	Jeu 06.06.19	22
24		★	Fin du TPI	0 jour	Jeu 06.06.19	Jeu 06.06.19	23

## 2 Analyse

### 2.1 Analyse concurrentielle

Les applications trouvées possèdent des limitations car l'analyse concurrentielle compare des solutions gratuites. Les interfaces graphiques peuvent également sembler basiques pour certains produits.

- VisuGPX: <https://www.visugpx.com/>
- uTrack: <http://utrack.crempa.net/>
- trackreport: <https://www.trackreport.net/>

## 2.1.1 uTrack

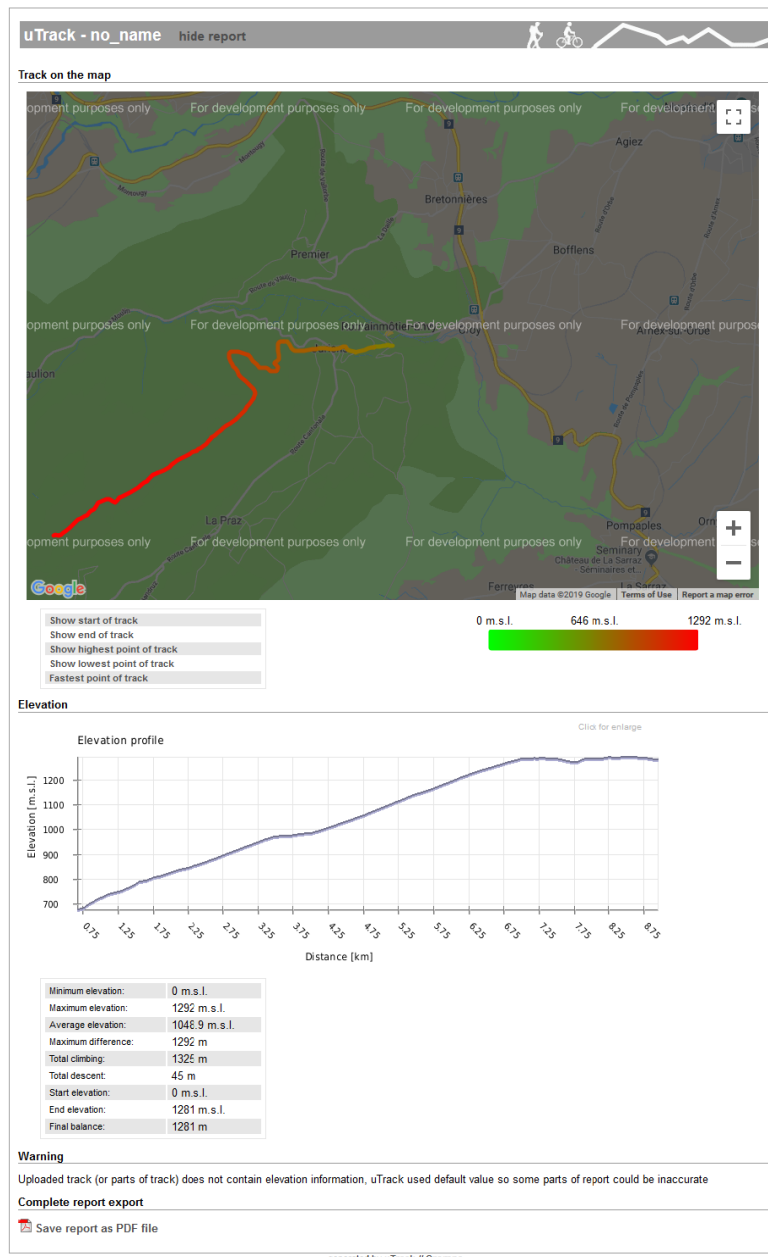
### Points positifs

- La carte est interactive

### Contraintes et limites

- La taille maximale des fichiers GPX pouvant être envoyés est de 1Mb
- Les graphiques sont représentés en tant qu'images, cela limite l'interaction que peut avoir un utilisateur avec ceux-ci.

### Interface graphique



La page propose des graphiques simple et les données de l'activité sont également affichées dans un tableau. Le dénivelé du parcours est affiché avec un gradient de couleur.

## 2.1.2 trackreport

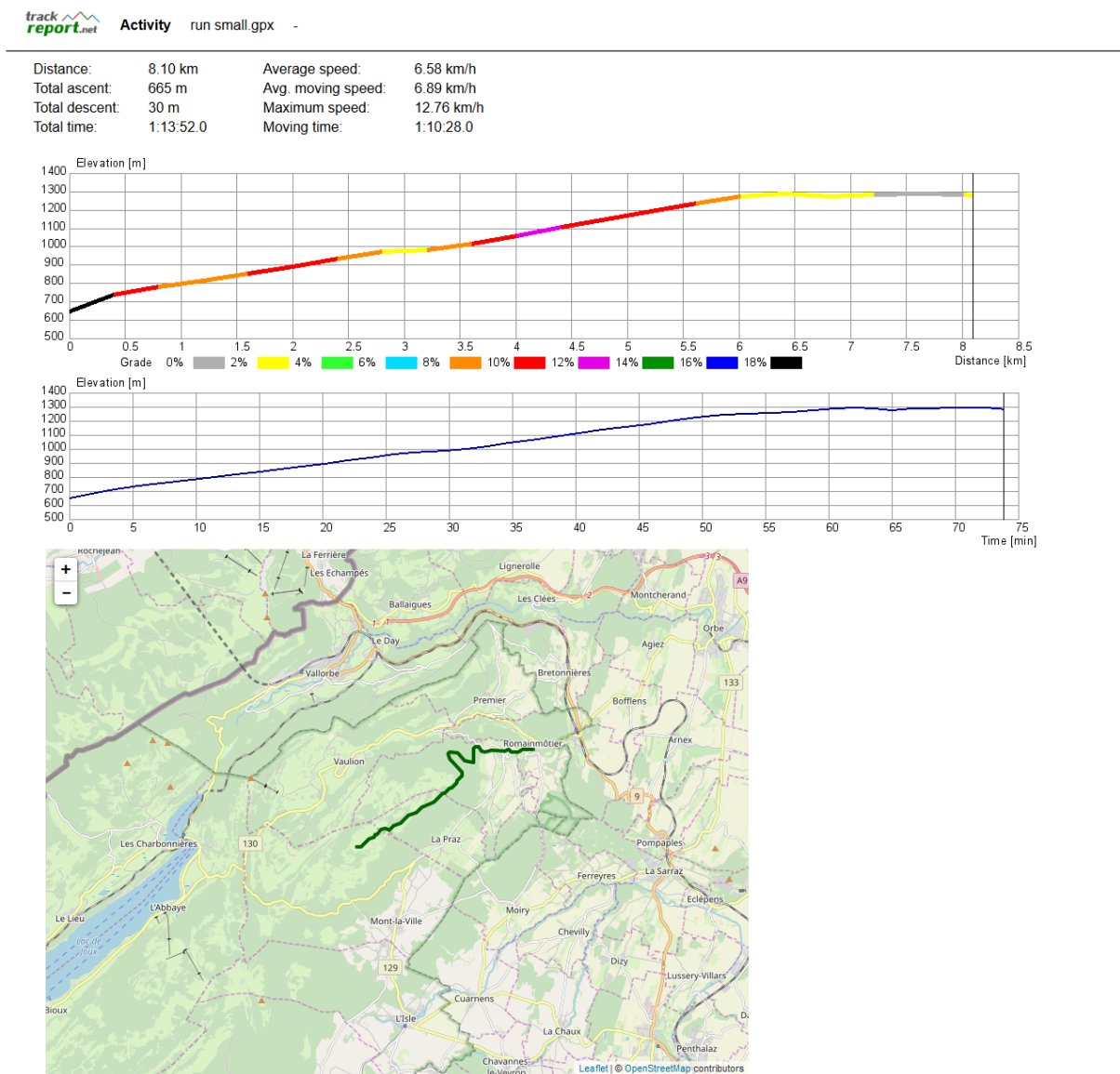
### Points positifs

- La carte est interactive

### Contraintes et limites

- La taille maximale des GPX pouvant être envoyés sont est de 2,5 Mb
- Les graphiques sont représentés en tant qu'images, cela limite l'interaction que peut avoir un utilisateur avec ceux-ci.

### Interface graphique



Generated by [trackreport.net](https://trackreport.net)

[Save as PDF](#)



## 2.1.3 VisuGPX

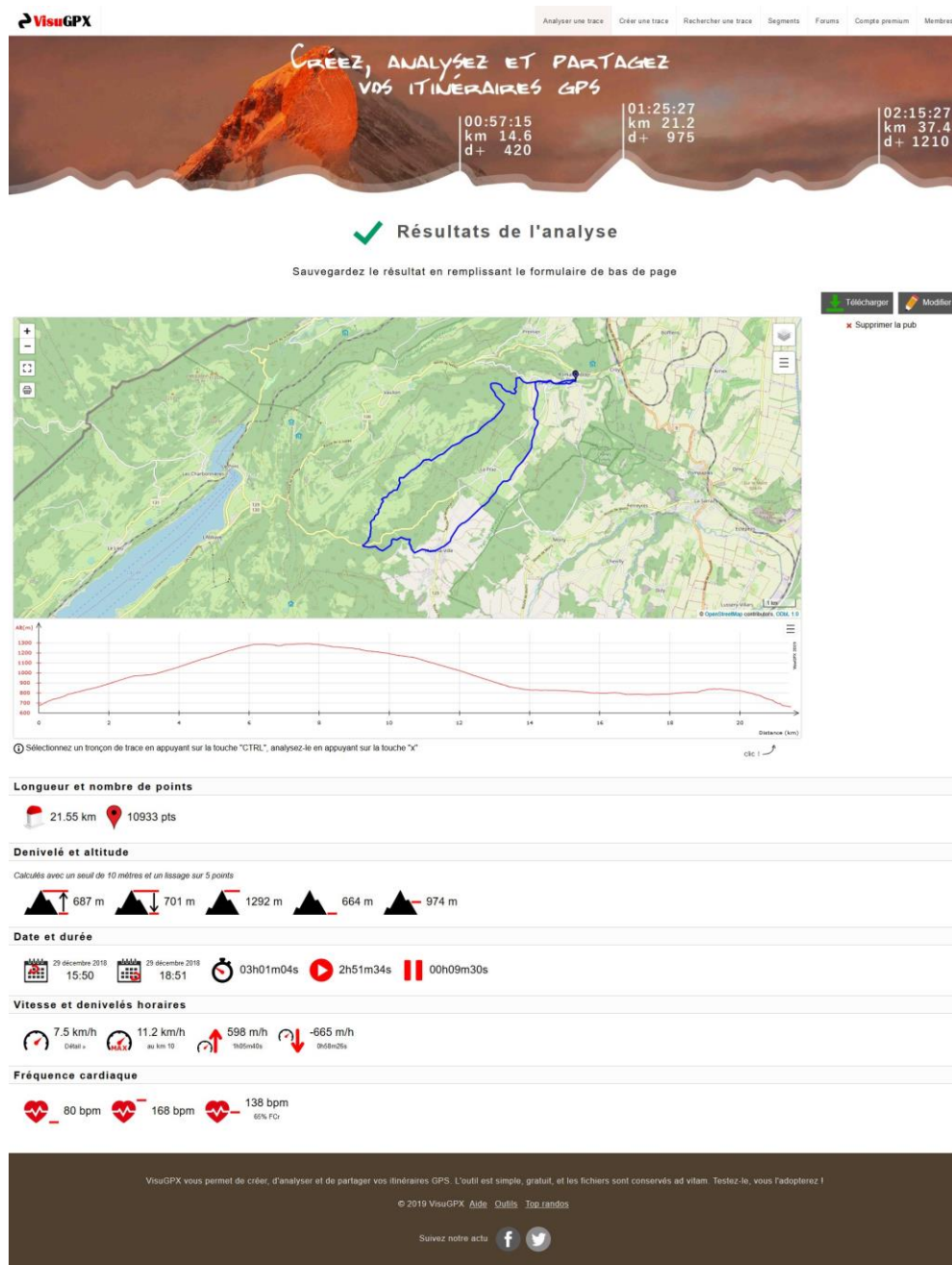
### Points positifs

- La carte et les graphiques sont interactifs

### Contraintes et limites

- La taille maximale des GPX pouvant être envoyés sont est de 2,5 Mb
- Les graphiques sont représentés en tant qu'images, cela limite l'interaction que peut avoir un utilisateur avec ceux-ci.

### Interface graphique



## 2.2 Création d'activités

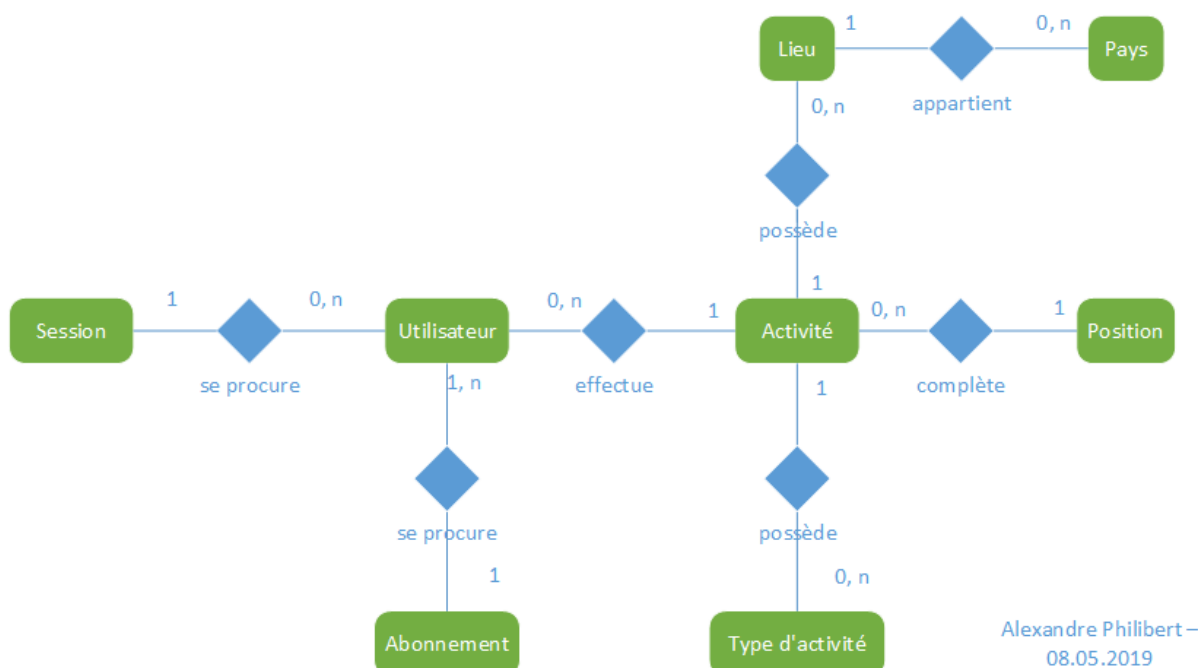
Le cahier des charges demande l'intégration de la création d'activités sportives. Le cahier des charges mentionne également la manipulation d'activités sportives sans et avec un fichier GPX<sup>1</sup>. L'application doit gérer le lieu et le pays de réalisation des activités, ces champs nécessitent d'être ajoutés à l'application.

Le sportif disposera d'une interface se présentant sous forme d'un client web lui permettant de créer des activités sportives. Pour faciliter la création d'une activité par un sportif, celui-ci aura la possibilité d'envoyer le fichier GPX entier à l'aide de l'interface graphique. Ce fichier pouvant s'avérer volumineux, des modifications doivent être apportées à l'application pour qu'elle puisse gérer le téléchargement des fichiers.

---

<sup>1</sup> Le GPX est un format de fichier qui permet d'enregistrer des positions GPS ainsi que des horodatages.

## Modèle conceptuel des données



## 2.3 Stratégie de test

### 2.3.1 Étendue des tests

Les tests seront effectués sur tous les endpoints de l'API. Des tests manuels et automatiques seront réalisés à l'aide du logiciel Postman.

### 2.3.2 Testeurs

- Alexandre Philibert : [alexandre.philibert@cpnv.ch](mailto:alexandre.philibert@cpnv.ch)

### 2.3.3 Type de tests

- Unitaire
- Intégration
- Fonctionnels

### 2.3.4 Liste des tests

- Use cases (décrit ci-dessous) à l'aide de Postman
- Scenarii (décrit ci-dessous) à l'aide de Postman
- Tests manuel de l'IHM

### 2.3.5 Données de test à prévoir

Des données aléatoires seront créées pour effectuer des tests sur l'API lors de la création d'utilisateurs et d'activités sans GPX. Des données provenant d'activités sportives fournies par Monsieur Glassey seront également utilisées.

## 2.4 Budget initial

Aucun budget n'est alloué à la réalisation de ce projet.

## 3 Conception

### 3.1 Ressources de l'API

Ressource	Utilisation
User	Stocke les données d'un utilisateur
Token	Créer un token d'authentification
Activity	Stocke une activité sportive réalisé par un sportif
Subscription	Stocke l'abonnement d'un utilisateur (date de début, date de fin, ...)
Position	Stocke une position GPS à un temps donné
Place	Contient les localités des pays ou le sportif peut réaliser une activité
Country	Contient la liste des pays ou le sportif peut réaliser une activité

### 3.2 Endpoints

Fonction	Endpoints	Verbe HTTP
Création compte	/user	POST
Modification compte	/user/{userid}	PUT
Authentification	/token	GET
Création d'activité	/user/{userid}/activity/	POST
Lecture d'activité	activity/{activityid}	GET
	/user/{userid}/activity/{activityid}	
Modification d'activité	activity/{activityid}	PUT
	/user/{userid}/activity/{activityid}	
Suppression d'activité	activity/{activityid}	DELETE
	/user/{userid}/activity/{activityid}	
Modification d'une position	/position/{positionid}	PUT
Suppression d'une position	/position/{positionid}	DELETE

Création type d'activité	/activity-type/	POST
Lecture types d'activité	/activity-type/	GET
Suppression type d'activité	/activity-type/{typeid}	DELETE
Modification type d'activité	/activity-type/{typeid}	PUT
Lecture des localités d'un pays	/country/{countryid}/place/	GET
Lecture des pays	/country	GET

### 3.3 Gestion lieu et pays

Le cahier des charges demande l'insertion de lieu et de pays de réalisation des activités sportives. Pour faciliter la création des activités pour le sportif, une liste des pays et des lieux seront accessible sur le endpoint country. Le cahier des charges ne demande pas la modification de la liste des lieux et pays par les utilisateurs, cette fonctionnalité ne sera donc pas intégrée.

Une liste contenant toutes les communes de Suisse a été récupérée :  
<https://www.bfs.admin.ch/bfs/fr/home/bases-statistiques/repertoire-officiel-communes-suisse.assetdetail.6986904.html>

## 3.4 Gestion des fichiers GPX

L'application dans son état actuel ne permet pas la gestion de fichier GPX trop volumineux (plusieurs Mb). Cela est dû au fait que le fichier GPX est envoyé dans le corps (body) de la requête HTTP d'une seule traite. Plusieurs possibilités existent pour résoudre ce problème :

### 3.4.1 Envoie de multiples requêtes

L'une des possibilités serait de découper le fichier GPX sur le client web et d'envoyer plusieurs requêtes consécutives contenant à chaque fois une partie du fichier. Cette méthode a comme désavantage de devoir renégocier le contexte d'envoi. Il serait aussi nécessaire de numérotter les parties de requêtes pour reconstruire le fichier dans l'ordre du côté l'API.

### 3.4.2 Requête "multipart/form-data"

Une autre possibilité est l'utilisation du *Content-Type multipart/form-data* du protocole HTTP. L'avantage de cette solution est la conservation du contexte de la requête, car une seule requête HTTP est envoyée au serveur contenant plusieurs parties. L'envoi de formulaires HTML est également simplifié avec cette solution, aucun traitement n'est nécessaire sur le client avant l'envoi du GPX.

#### **Formidable**

Le module node Formidable permet de gérer les requêtes *multipart/form-data*. Ce module stocke néanmoins le fichier sur le serveur (dans le répertoire temporaire par défaut). Il serait donc nécessaire d'ajouter une gestion plus approfondie de ces fichiers.

#### **Multer**

Le module node Multer permet également de gérer les requêtes *multipart/form-data*. L'avantage de ce module est de pouvoir stocker le fichier uniquement en mémoire vive, cette fonctionnalité n'est cependant pas nécessaire si nous souhaitons stocker le fichier sur le serveur.

#### **Solution retenue**

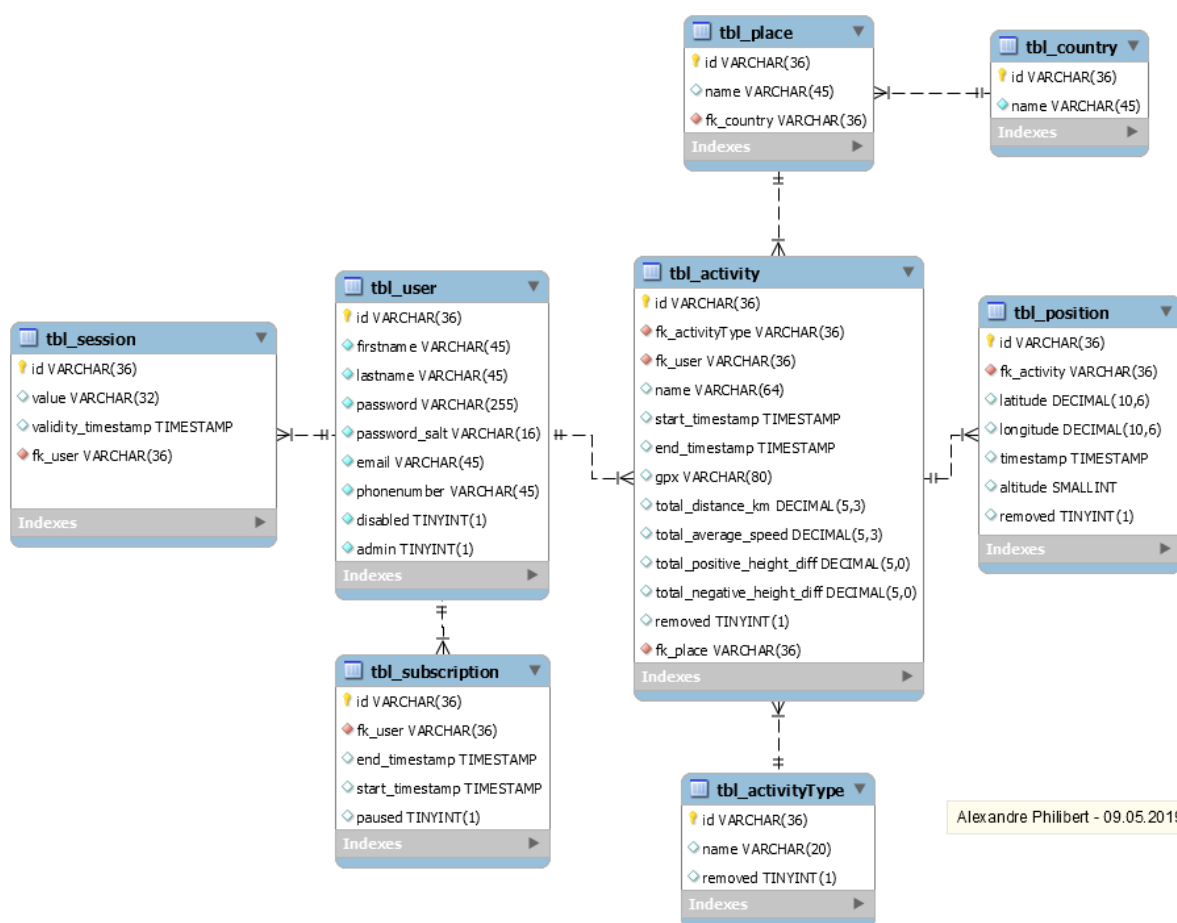
Le module Formidable semble, dans ce cas, plus adapté. Ce module n'intègre pas le stockage fichier en mémoire vive, mais il ne semble pas nécessaire d'intégrer cette fonctionnalité. La syntaxe du module semble aussi plus simple que Mutler

## 3.5 Intégration du middleware Formidable

Un middleware doit être enregistré dans le routeur principal (src/router.js) pour gérer les requêtes *multipart/form-data* reçue par l'API. Le middleware sera enregistré dans le répertoire src/middlewares.

Il est nécessaire de modifier la table `tbl_activity`, le champs GPX stockant actuellement le fichier dans son intégralité devra désormais stocker le chemin vers le fichier dans le système de fichier de l'OS.

## 3.6 Modèle logique des données



Alexandre Philibert - 09.05.2019

### 3.6.1 tbl\_activity

Champs	Utilisation
gpx	chemin jusqu'au fichier GPX stocké sur le serveur
total_positive_height_diff	Dénivelé positif total exprimé en valeur positive
total_negative_height_diff	Dénivelé négatif total exprimé en valeur positive

### 3.6.2 tbl\_country

Champs	Utilisation
name	Nom du pays

### 3.6.3 tbl\_place

Champs	Utilisation
name	Nom de la localité

## 3.7 Client web

Plusieurs solutions de serveurs HTTP existent, la solution mise en œuvre actuellement sur l'application est assez basique en NodeJS. Lorsqu'un client demande un page web, l'application fait une correspondance entre le nom demandé est un nom de fichier sur le serveur, l'application va ensuite lire le fichier et l'envoyer au client. Le fichier est lu à chaque nouvelle requête, aucun cache n'est présentement en place. Les contraintes de temps ne permettent pas de mettre en place un système plus avancé que celui-ci.

L'architecture du client va cependant être modifiée pour permettre aux sportifs d'avoir une meilleure expérience utilisateur.

### 3.7.1 Architecture client

#### **Pages statiques et appels XHR**

Cette solution est actuellement mise en place sur l'application. Elle est la plus simple à mettre en œuvre car elle ne nécessite pas de traitements particuliers. Des appels XHR sont cependant nécessaires pour personnaliser le contenu des pages tel que l'affichage des activités appartenant à un sportif.

Cette solution ne permet pas de simplement réutiliser des parties du DOM tel que le menu par exemple.

#### **Génération des pages en PHP**

La génération des pages en PHP peut nécessiter la mise en place d'un serveur HTTP tel qu'Apache ou NGINX. Cette solution génère les pages servies au client sur le serveur, il est donc possible de réutiliser des parties du DOM en utilisant des designs patterns tel que le MVC.

#### **Génération de pages en Javascript (coté client)**

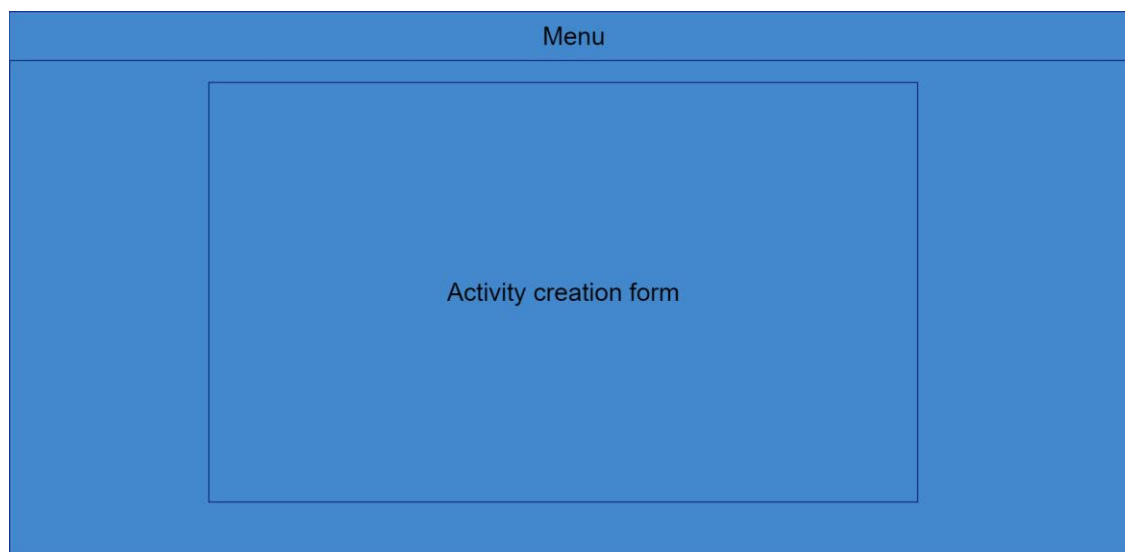
La génération du DOM sur le client permet de diminuer la charge du serveur. Il est aussi possible grâce à cette solution de garder le contexte d'exécution car aucun rechargement de pages n'est nécessaire.

Cette solution a été choisie car elle permet de garder une application développée uniquement en Javascript autant du côté back-end que front-end.



## 3.7.2 Création d'activité – sans GPX (Story 001)

### Zonning



## Wireframe

The wireframe shows a web browser window titled "Runscape" with the URL "http://runscape.ch/create-activity". The page has a header with a "Logo" placeholder and a user name "John Doe". The main content area contains a form for creating an activity with the following fields:

- Nom:** Text input with value "Course à pied à Ottenbach".
- Lieu:** Two dropdown menus, the first with "Suisse" and the second with "Yverdon-les-Bains".
- Type:** Dropdown menu with "Course à pied".
- Début:** Datetime input with value "2018-06-12T15:30".
- Fin:** Datetime input with value "2018-06-12T19:30".
- Distance:** Text input with value "212" and unit "km".
- Dénivelé positif:** Text input with value "200" and unit "m".
- Dénivelé négatif:** Text input with value "100" and unit "m".
- Vitesse moyenne:** Text input with value "7.45" and unit "km/h".

Below the form, there is a "Message d'erreur" label and a "Créer" button.

Champ	Type (HTML)	Commentaires
Nom	text	
Lieu	select / select	Les champs Pays et ville sont regroupé sur la même ligne
Type	select	Les données sont récupérées depuis le endpoint /activity-type
Début	datetime-local	Ce type de champs n'est pas supporté par firefox, ie ainsi que safari, il est cependant le seul champ ne nécessitant pas de vérification supplémentaire pour validé son contenu.
Fin	datetime-local	Idem que le champ Début
Distance	number	
Dénivelé pos.	number	
Dénivelé neg.	number	
Vitesse moy.	number	

Des contraintes de valeurs minimales et maximales seront ajoutés sur certains champs.

## 3.8 Cas d'utilisations

### 3.8.1 Sportif

En tant que sportif	Je veux enregistrer une activité sportive sans GPX	Dans le but de suivre ma progression
Déroulement	Étape	Description
	1	Membre : Soumet une requête POST à "user/{userid}/activity"
	2	API : Valide la requête
	3	API : Valide l'autorisation de création de la ressource
	4	API : enregistre les informations dans la base de données
	5	API : Retourne l'état de l'enregistrement au membre
Extensions	2a	Autorisation non valide API : Retourne une erreur d'autorisation
	3a	Format de date de début invalide API : Retourne une erreur de format de date
	3b	Format de date de fin invalide API : Retourne une erreur de format de date
	3c	ID de type d'activité inexistant API : Retourne une erreur de format d'activité
	3d	Champs fichier GPX vide, autres champs vides API : Retourne une erreur de champs manquant

En tant que sportif	Je veux enregistrer une activité sportive avec GPX	Dans le but de suivre ma progression
Déroulement	Étape	Description
	1	Membre : Soumet une requête POST à "user/{userid}/activity"
	2	API : Valide la requête
	3	API : Valide l'autorisation de création de la ressource
	4	API : enregistre les informations dans la base de données
	5	API : Retourne l'état de l'enregistrement au membre

<b>Extensions</b>	2a	Autorisation non valide API : Retourne une erreur d'autorisation
	2b	Format de GPX invalide API : Retourne une erreur de format GPX invalide
	3a	Format de date de début invalide API : Retourne une erreur de format de date
	3b	Format de date de fin invalide API : Retourne une erreur de format de date
	3c	ID de type d'activité inexistant API : Retourne une erreur de format d'activité
	3d	Champs fichier GPX vide, autres champs vides API : Retourne une erreur de champs manquant

En tant que sportif	Je veux consulter la liste des pays	Dans le but de choisir le pays où j'ai réalisé mon activité
---------------------	-------------------------------------	---

Déroulement	Étape	Description
	1	Membre : Soumet une requête POST à "country"
	2	API : Valide la requête
	5	API : Retourne la liste des pays

En tant que sportif	Je veux consulter la liste des localité d'un pays	Dans le but de choisir la localité où j'ai réalisé mon activité
---------------------	---	---

Déroulement	Étape	Description
	1	Membre : Soumet une requête POST à "country/{countryid}/place"
	2	API : Valide la requête
	5	API : Retourne la liste des localités du pays
<b>Extensions</b>	2a	L'identifiant du pays (countryid) n'existe pas API : Retourne une liste vide

## 4 Réalisation

### 4.1 Description des tests effectués

En tant que sportif	Je veux consulter la liste des pays	Dans le but de choisir le pays où j'ai réalisé mon activité
---------------------	-------------------------------------	---

Déroulement	Étape	Description	Résultat
	1	Membre : Soumet une requête POST à "country"	
	2	API : Valide la requête	
	5	API : Retourne la liste des pays	Réussi

En tant que sportif	Je veux consulter la liste des localité d'un pays	Dans le but de choisir la localité où j'ai réalisé mon activité
---------------------	---	---

Déroulement	Étape	Description	Résultat
	1	Membre : Soumet une requête POST à "country/{countryid}/place"	
	2	API : Valide la requête	
	5	API : Retourne la liste des localités du pays	Réussi
<b>Extensions</b>	2a	L'identifiant du pays (countryid) n'existe pas API : Retourne une liste vide	Réussi

## 5 Annexes

### 5.1 Sources

HTTP POST multipart/form-data

<https://developer.mozilla.org/fr/docs/Web/HTTP/M%C3%A9thode/POST>

Utilisation des objets FormData :

[https://developer.mozilla.org/fr/docs/Web/API/FormData/Utilisation\\_objets\\_FormData](https://developer.mozilla.org/fr/docs/Web/API/FormData/Utilisation_objets_FormData)

Liste communes Suisse

<https://www.bfs.admin.ch/bfs/fr/home/bases-statistiques/repertoire-officiel-communes-suisse.assetdetail.6986904.html>

Sélection d'une ligne sur N MySQL

<https://stackoverflow.com/questions/858746/how-do-you-select-every-n-th-row-from-mysql>

datetime-local

[https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/datetime-local#Browser\\_compatibility](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/datetime-local#Browser_compatibility)