

# TPI – Shoot4Stats

---

« SHOOT4STATS » APPLICATION WEB DE GESTION  
DE TIR À L'ARC

# Contents

<b>CONTENTS</b>	<b>1</b>
<b>1 ANALYSE PRÉLIMINAIRE</b>	<b>2</b>
1.1 INTRODUCTION.....	2
1.2 OBJECTIFS.....	2
1.3 CONTEXTE .....	2
1.3.1 STRUCTURE D'UN <i>SHOOT</i>	2
1.3.2 REMPLISSAGE D'UN <i>SHOOT</i>	2
<b>2 ANALYSE</b>	<b>3</b>
2.1 CAHIER DES CHARGES DÉTAILLÉ .....	3
2.1.1 PRÉREQUIS	3
2.1.2 DESCRIPTIF DU PROJET	3
2.1.3 LOGICIELS ET RESSOURCES À DISPOSITION	3
2.2 STRATÉGIE DE TEST.....	3
2.2.1 STRATÉGIE GLOBALE	3
2.2.2 PROCÉDURE	4
2.3 ETUDE DE FAISABILITÉ .....	4
2.3.1 RISQUES TECHNIQUES	4
2.3.2 RISQUES SUR LE PLANNING	4
2.4 PLANIFICATION .....	4
<b>3 DOSSIER DE CONCEPTION</b>	<b>5</b>
3.1 PROCESSUS POUR ARCHERS .....	5
3.2 BASE DE DONNÉES.....	5
3.2.1 MLD	5
3.2.2 DICTIONNAIRE DE DONNÉES	5
3.3 API .....	7
3.4 HISTORIQUE.....	7
<b>4 RÉALISATION</b>	<b>8</b>
4.1 DOSSIER DE RÉALISATION .....	8
4.1.1 MISE EN PLACE DU PROJET	8
4.2 DESCRIPTION DES TESTS EFFECTUÉS.....	8
4.3 ERREURS RESTANTES .....	8
4.4 DOSSIER D'ARCHIVAGE.....	8
<b>5 MISE EN SERVICE</b>	<b>9</b>
5.1 RAPPORT DE MISE EN SERVICE .....	9
5.2 LISTE DES DOCUMENTS FOURNIS.....	9
<b>6 CONCLUSIONS</b>	<b>9</b>
<b>7 ANNEXES</b>	<b>10</b>
7.1 SOURCES – BIBLIOGRAPHIE.....	10
7.2 JOURNAL DE TRAVAIL .....	10
7.3 MANUEL D'INSTALLATION .....	10
7.4 MANUEL D'UTILISATION .....	10
7.5 ARCHIVES DU PROJET.....	10

## 1 Analyse préliminaire

### 1.1 Introduction

Mon travail de fin d'apprentissage a pour but de rentrer un Shoot (tir à l'arc) dans la DB afin d'avoir des statistiques rapidement sur le Shoot.

Ce travail permet de fortifier mes connaissances en développement WEB et d'apprendre et voir comment se déroule le développement d'une application WEB de manière professionnelle.

Le choix de cette application a été fait en corrélation avec le chef de projet à ma demande, pour répondre à des besoins personnels et peut-être même aux besoins des archers en règle générale.

### 1.2 Objectifs

Développer une interface WEB « frontend » et mobile first puis adapter le « backend » si besoin afin d'effectuer des tâches pour un utilisateur avec un rôle propre au sein de l'application.

### 1.3 Contexte

#### 1.3.1 Structure d'un Shoot

Un Shoot est défini comme suit :

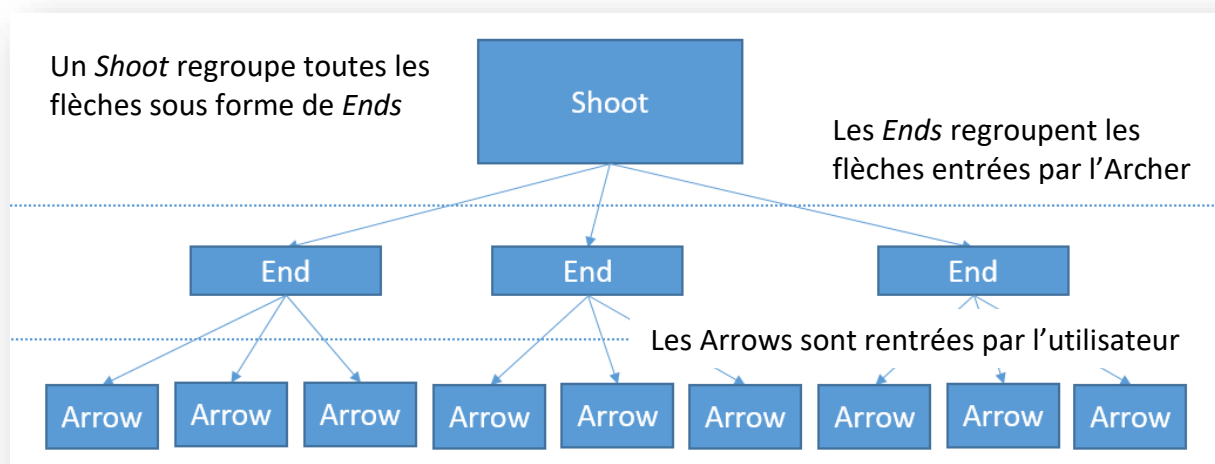


Figure 1 Schéma d'un Shoot

#### 1.3.2 Remplissage d'un Shoot

La séquence de tir au tir à l'arc se passe comme suit ;

- L'on tire n flèches (*Arrows*) 3 pour les compétitions **Indoor** et 6 pour les compétitions **Outdoor**
- Ces n flèches se « regroupent » et forment une volée « *End* »
- Et après n volées (20 pour **Indoor**, 12 pour **Outdoor**) Le Shoot se termine, sachant que généralement l'on marque un point d'arrêt à la moitié du Shoot afin de définir deux séries.

## 2 Analyse

### 2.1 Cahier des charges détaillé

#### 2.1.1 Prérequis

Une partie du travail a déjà été réalisé en collaboration avec le chef de projet, soit :

- Modéliser, implémenter et documenter la base de données
- Mettre en place un environnement de développement
  - IDE (en l'occurrence Visual Studio Code)
  - Gestion de source (GIT)
- Développer la partie « Backend » sous la forme d'une API
  - Requêtes principales préalablement développée
  - Authentification externe via Facebook à l'aide de PassportJS

#### 2.1.2 Descriptif du projet

- Les Archers « utilisateurs authentifiés » doivent pouvoir ajouter des Shoots complets et voir les détails de ceux-ci. Reprendre les Shoots s'ils n'ont pas été terminés.
- Aux « administrateurs » (définis au sein de la DB directement) doivent pouvoir lister les Archers et avoir accès à leurs « Shoots ».
- Aux utilisateurs anonymes (non authentifiés) de voir les statistiques du dernier Shoot d'un utilisateur en ayant l'URL personnelle.

#### 2.1.3 Logiciels et ressources à disposition

- Un PC :
  - OS : Windows 10 Etudiant
  - IDE : Visual Studio Code
  - Langage : JavaScript
  - Framework : NodeJS **v.7.4.0** (API)
  - Librairies : VueJS (FrontEnd) / Axios / JQuery / Framework CSS Materialize / ExpressJS (BackEnd) / ORM Sequelize / PassportJS gère l'authentification
- Services :
  - Authentification externe via Facebook Login

### 2.2 Stratégie de test

#### 2.2.1 Stratégie globale

Comme le cahier des charges le spécifie la méthode de développement utilisée doit être itérative (AGILE), voici donc les 4 valeurs fondamentales de la méthode AGILE :

- **L'équipe** ("Personnes et interaction plutôt que processus et outils")
- **L'application** ("Logiciel fonctionnel plutôt que documentation complète")
- **La collaboration** ("Collaboration avec le client plutôt que négociation de contrat")
- **L'acceptation du changement** ("Réagir au changement plutôt que suivre un plan")

L'on remarque que l'accent est mis sur le client, le faire participer un maximum au développement de l'application afin qu'il se sentes concerné et que le produit fini lui convienne au maximum au lieu de se baser uniquement sur un cahier des charges fixe.

### 2.2.2 Procédure

Lors des gros ajouts de fonctionnalités je ferai tester à des personnes externes (archers, coach de tir) dans la limite du possible afin d'avoir un retour externe sur les tests (interface utilisateur et simplicité d'utilisation)

Puis de mon côté lors du développement j'effectuerai des tests tels que :

1. Explication du test
2. Résultats attendus
3. Résultats observés
4. Discussion des résultats / résolution des problèmes

Mes tests s'effectueront sur la validation des formulaires, la restriction relative aux pages de l'application, la récupération des informations sur l'API.

## 2.3 Etude de faisabilité

---

### 2.3.1 Risques techniques

La complexité technique réside dans la modularisation du code afin de permettre une bonne évolutivité du code pour permettre les améliorations / ajouts de fonctionnalités futures de fonctionner rapidement grâce aux modules présents.

Ayant déjà fait le BackEnd en collaboration avec le chef de projet, ce concept de modularité a été appris à ce moment-là, je vais donc faire en sorte de bien séparer mes Vues et que les données soient bien partagées par composants afin de toujours savoir où l'on en est.

### 2.3.2 Risques sur le planning

Je pense qu'au niveau du planning il ne va pas y avoir des gros problèmes tant que je ne m'éloigne pas de l'objectif principal (c'est-à-dire d'avoir une application fonctionnelle avant tout) et que je me concentre sur les points principaux.

Pour cela j'ai défini des priorités sur les tâches comme on peut le voir dans le Planning Initial, je ferai d'abord la partie « Archers », puis Administrateurs et enfin les utilisateurs anonymes.

## 2.4 Planification

---

Voir en annexe « Planning Initial ».

### 3 Dossier de conception

#### 3.1 Processus pour Archers

Processus de création de Shoot et remplissage de celui-ci

#### 3.2 Base de données

La base de données fait partie des prérequis du projet, c'est-à-dire qu'elle a été faite en collaboration avec le chef de projet soit Stefano Nepa.

##### 3.2.1 MLD

Le MLD a été généré par MySQL WorkBench d'après le diagramme que j'ai fait.

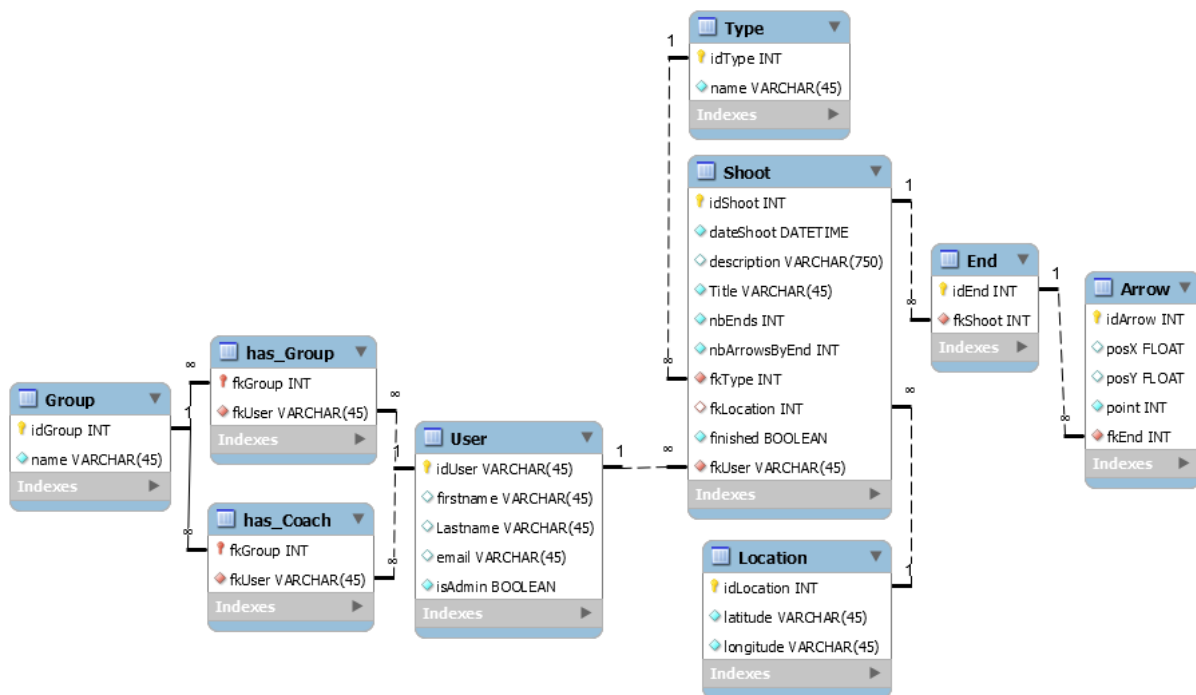


Figure 2 Modèle Logique des données

##### 3.2.2 Dictionnaire de données

Voici la description de la base de données :

- **Shoot**  
Représente un Shoot.
  - **idShoot**  
Id du Shoot, auto incrémentiel, permet de pouvoir créer des Shoots sans contraintes
  - **dateShoot, description, title**  
Propriétés du Shoot, définition du Shoot  
dateShoot : valeur par défaut CURRENT\_TIMESTAMP  
Description et title set par l'utilisateur
  - **nbEnds, nbArrowsByEnd**  
L'API chargera automatiquement suivant le type du Shoot, sauf si le type est « Training » auquel cas ce sera à l'utilisateur de choisir ses chiffres
  - **fkType**  
Clé étrangère pour la définition du Type du Shoot

- **fkLocation**  
Clé étrangère qui peut être nulle, définit la *location* du Shoot si l'utilisateur le veut
- **fkUser**  
Clé étrangère définissant l'Archer qui a tiré le Shoot
- **finished**  
Propriété qui est définie à « true » par l'API quand le nombre de *Ends* associée avec le Shoot est le même que le nombre de *Ends* mise dans **nbEnds**
- **Type**  
Représente le Type du Shoot
  - **idType**  
Identifiant du type
  - **name**  
Le nom du Type, dans ce cas il y en a 3
- **Location**  
Permet d'ajouter une localisation au *Shoot*
  - **idLocation**  
Identifiant unique de la localisation (peut-être redondance des données si des utilisateurs tirent au même endroit)
  - **latitude, longitude**  
Description de la localisation, dans le cas de mon application je pensais utiliser les services de Google Maps, si le temps me manque la ville et la rue Seront dans ces 2 champs en clair → à corriger si cela se produit
- **End**  
Contient les flèches tirées, appartient au *Shoot*
  - **idEnd**  
Permet l'identification au sein des Arrows, agit uniquement comme une table intermédiaire « Conteneur » Ajoute de la logique à la DB
  - **fkShoot**  
Décrit l'appartenance à un *Shoot*
- **Arrow**  
Décrit la flèche et sa valeur tirée par l'archer
  - **idArrow**
  - **posX, posY, point**  
Décrit la position de la flèche sur la cible (hors cadre du TPI mais possibilités d'amélioration), et le point qui est attribué à la flèche (attribut le plus important de la DB)
  - **fkEnd**  
Détermine sur quel End la flèche va venir se greffer afin de respecter ce principe de séries et de chronologie évoqué plus haut.
- **User**  
Décrit un utilisateur qu'il soit admin ou archer
  - **idUser**  
Id unique attribué par Facebook<sup>1</sup> pour l'user utilisant cette application, je m'en sers afin d'identifier les utilisateurs, lors de l'entrée dans la DB l'ID est « salté » avec *facebook\_*

---

<sup>1</sup> Pour plus de détails : <https://developers.facebook.com/docs/graph-api/reference/v2.5/user>

- **firstname, lastname, email**  
Informations relatives à l'utilisateur, récupérées via Facebook
- **isAdmin**  
Propriété booléenne afin de mettre un utilisateur en « Admin »
- **has\_Group, has\_Coach, Group**  
Ces tables ne seront pas utilisées dans le cadre de ce TPI mais seront utilisées ensuite afin de définir des rôles bien plus précis aux utilisateurs

### 3.3 API

---

Voir Annexes

<https://github.com/mbonjour/Shoot4Stats/blob/master/Documentation/Annexes/API/apiDocumentation.md> et

<https://github.com/mbonjour/Shoot4Stats/blob/master/Documentation/Annexes/API/authDocumentation.md>

### 3.4 Historique

---

Je listerai au mieux les changements effectués, sachant que ce sera une programmation suivant la norme « AGILE » qui consiste à discuter en permanence avec le client et à développer par étapes l'application. Je vais donc répertorier ici les discussions avec le chef de projet / client.

*Si la conception du projet a dû être modifiée plusieurs fois, ou de manière significative, expliquez ces changements et leurs causes.*

*Attention: Pour faciliter la maintenance, à la fin du projet, le dossier de conception doit correspondre à ce qui a été effectivement réalisé !*



## 4 Réalisation

### 4.1 Dossier de réalisation

#### 4.1.1 Mise en place du projet

Explications : Vue init – Vue CLI

Choix du boilerplate, « webpack » → Pourquoi Webpack et qu'est-ce que c'est en bref

### 4.2 Description des tests effectués

*Pour chaque partie testée de votre projet, il faut décrire:*

- *les conditions exactes de chaque test*
- *les preuves de test (papier ou fichier)*
- *tests sans preuve: fournir au moins une description*

### 4.3 Erreurs restantes

*S'il reste encore des erreurs:*

- *Description détaillée*
- *Conséquences sur l'utilisation du produit*
- *Actions envisagées ou possibles*

### 4.4 Dossier d'archivage

*Décrire de manière détaillée les 2 archives du projet (CD-ROM, disque zip ou jazz, bandes magnétiques, ...)*

**Attention: les documents de réalisation doivent permettre à une autre personne de maintenir et modifier votre projet sans votre aide !**

## 5 Mise en service

### 5.1 Rapport de mise en service

*Fournir une description:*

- *de l'installation du projet chez le client (pour un site web: publication chez un provider)*
- *des test officiels effectués chez le client et/ou par le client.*
- *des erreurs répertoriées*
  - *description détaillée*
  - *conséquences pour le client*
  - *actions envisagées.*

### 5.2 Liste des documents fournis

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

- *le rapport de projet*
- *le manuel d'Installation (en annexe)*
- *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
- *autres...*

## 6 Conclusions

*Développez en tous cas les points suivants:*

- *Objectifs atteints / non-atteints*
- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

## 7 Annexes

### 7.1 Sources – Bibliographie

---

*Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)... Et de toutes les aides externes (noms)*

### 7.2 Journal de travail

---

### 7.3 Manuel d'Installation

---

### 7.4 Manuel d'Utilisation

---

### 7.5 Archives du projet

---

*CD, ... dans une fourre en plastique*

# Annexes