

# Spécifications projet

## Logitud

## Championnat de ski

## **Contexte :**

Depuis 11 ans, Logitud sponsorise les championnats de ski de la police municipale. Les polices sont rattachées aux communes. Une association gère cette compétition. 100-150 personnes tous les ans. C'est un championnat dont l'encadrement est réalisé par les membres de l'ESF. Ce championnat rassemble entre 100 et 150 personnes tous les ans.

Nous souhaitons leur proposer un outil permettant de générer un document excel a compléter lors de la compétition. Cet outil aura également la tache de gérer l'affichage des résultats.

# Spécification techniques:

## ***Processus de création des données:***

Au départ l'application doit permettre de créer une épreuve, et d'en saisir les compétiteurs. Quand l'intégralité des participants est saisi, l'application pourra permettre d'exporter un fichier CSV pré-rempli.

L'interface devra permettre d'exporter un fichier CSV prérempli, qui pourra être complété par les organisateurs. Une fois la compétition terminée ce fichier devra être réimporté dans l'application et intégrer les données dans une base de données.

L'application ne contiendra donc pas de formulaire permettant la saisie des temps en direct.

Dans le cas d'un nouvel envoi d'un fichier, les données précédemment saisies sont écrasées.

## Contraintes métier et définitions:

Vous devrez créer une base de données permettant de stocker les données organisées comme suit.

### ***Epreuve:***

Chaque année est donc organisée une épreuve, faisant participer de nombreuses personnes.

Un épreuve est définie comme suit:

- Lieu : chaque année l'épreuve se déroule dans un autre endroit
- Date: la date a laquelle l'épreuve se déroule
- Participants: liste des personnes qui prennent part a l'épreuve

### ***Participant:***

Un participant est une personne qui prend part a la compétition:

Chaque participant est défini comme suit:

- Nom
- Prénom
- Date de naissance
- Catégorie
- Adresse email
- Profil
- Photo

### ***Catégorie:***

Un participant participe a l'épreuve en étant inscrit dans une catégorie. Cette liste de catégorie est prédéfinie:

- M1
- M2
- M3
- Sénior
- V
- Snow
- Nouvelle Glisse(NG)

Ces catégories sont susceptibles d'évoluer. Un participant ne peut participer que dans une catégorie.

### ***Profil:***

Le profil est une seconde sorte de catégorisation. Par défaut les participants sont tous des policiers municipaux, cependant il existe d'autres profils.

- ASVP : ex-pervenches
- Open: conjoints et/ou enfants
- Gardes-champêtres

Ces catégories sont susceptibles d'évoluer. Un participant ne peut avoir qu'un seul profil.

### ***Passage:***

Chaque concurrent qui participe à l'épreuve, a droit à deux passages (descentes). Chaque passage est donc chronométré. Chaque passage devra également comporter une information qui permet de l'identifier comme premier ou second passage.

### ***Résultat:***

Le résultat de chaque participant est composé de la moyenne de ses deux passages.

## **Implementation technique backend:**

Le projet devra être écrit en **PHP orienté objet et les dépendances seront gérées via composer**. Il sera possible d'utiliser les composants Symfony. L'utilisation du framework en lui-même n'est pas demandée. L'usage de doctrine n'est pas non plus souhaité.

La version de PHP utilisée sera la 7.4 afin de permettre de typer fortement un maximum d'éléments logiciels.

Les composants suivants pourront être utilisés:

- HttpFoundation
- Serializer
- Twig
- Carbon

### ***Qualité de code et fiabilisation***

Les classes PHP devront faire l'objet de tests unitaires écrits avec la librairie PHPUnit. L'usage d'autres Framework de tests est possible (Behat, PestPHP...)

L'application sera fournie avec un script SQL permettant de créer et remplir une base de données en vue des tests.

## Besoins fonctionnels de l'application:

Une fois le fichier complété et importé dans l'application par les gestionnaires de l'épreuve, l'application devra permettre d'afficher les résultats sous diverses formes.

- **Classement général:** toutes catégories confondues, le classement est complet
- **Classement par catégories:** pour chaque catégorie, seuls les 3 premiers sont affichés
- **Classement par catégories d'âge:** pour chaque tranche d'affichage des 3 premiers seulement

Le classement devra présenter la photo de chaque compétiteur à côté de son classement.

## Charte graphique:

Une charte graphique doit également être produite par le développeur.

L'application sera donc marquée comme produit « Logitud »

L'application devra être désignée avec un outil tel que Figma ou Adobe XD et la marque Logitud devra être clairement identifiable.

Le maquettage pourra se faire sous forme de Wireframe.



# **Implémentation technique de la charte graphique:**

La création des pages sera faite en HTML. Pour la mise en forme, un framework du type « Bootstrap » peut être utilisé.

La partie CSS devra être faite en utilisant SCSS/SASS, afin de conserver une cohérence dans l'interface, ainsi que dans les différents composants.

Pour pouvoir utiliser SCSS/SASS, il sera essentiel d'utiliser Npm ou Yarn afin de gérer les dépendances javascript.

## **Déroulement du projet**

Jour1 & 2 :

Création des wireframes et conception visuelle de l'application. Détailler les scénarios possibles. Succès, erreurs , retours d'information a l'utilisateur.

Jours 3,4,5 : Modélisation de la base de données.

Ensuite début de la. programmation.