



AUNAY VTT

# APPLICATION ENDURO

DOSSIER DE SOUTENANCE





AUNAY VTT

# APPLICATION ENDURO

# PLAN

## 1- PRÉAMBULE

A- PRÉSENTATION GLOBAL DU PROJET	page 6
B- LISTE DES COMPÉTENCES COUVERTES PAR LE PROJET	page 7

## 2- CONTEXTE DU PROJET

A- PRÉSENTATION DU CAPB	
1- Qu'est ce que c'est ? De quoi s'agit-il ?	page 9
2- L'organisation au sein de l'association.	page 10
B- LE BESOIN	
1- Le cahier des charges	page 12
C- L'ORGANISATION	
1- Organisation des tâches	page 15
2- Le client au coeur de la conception	page 16



AUNAY VTT

# APPLICATION ENDURO

# PLAN

## 3- CONCEPTION ET DEVELOPPEMENT DU PROJET

### A- SPÉCIFICATIONS TECHNIQUES

- 1- Les technologies page 18
- 2- La sécurité page 20

### B- BASE DE DONNÉES

- 1- MCD et relations page 21
- 2- Dictionnaire de données page 20
- 3- Intégration à l'application page 25

### C- L'APPLICATION MOBILE

- 1- L'arborescence page 31
- 2- Logique métier et objet java page 33
- 3- Bonnes pratiques et documentation page 35
- 4- Maquettes UX page 37

### D- L'API page 40

### E- PHASE DE TEST page 42



AUNAY VTT

# APPLICATION ENDURO

# PLAN

## 4- CONCLUSION

A- ERREURS ET RÉOLUTIONS	page 44
B- ÉVOLUTIONS FUTURES ENVISAGÉES	page 45
C- AVANT / APRÈS	page 46
D-TRAVAIL DE VEILLE	page 49



AUNAY VTT

# APPLICATION ENDURO

# 1- PRÉAMBULE





AUNAY VTT

# APPLICATION ENDURO

## A- PRESENTATION OF THE ENDURO PROJECT

The concept of the enduro application come from the client who is the association Aunay VTT.

The goal is to facilitate the organisation of races. To do it they ask an application who permitted to automate and simplify the management process for the enduro.

The enduro is a kind of races, but some part of the course are timed. When runner comes at a part where they have to be, an agent say the top and then the runner start to run.

This project had a lot of trade constraint so object were the best way to manage it.

One of the constraints was that agent don't have any connexion or services during the race but the application have to centrilazed data in a extern database. So we need an API so agent can access the global result when they refound connexion.

So the project contain the mobile application, one API and an extern database. The mobile application was develop with android studio in the programming language java. The API was develop with symfony in php.



AUNAY VTT

# APPLICATION ENDURO

## B- LISTE DES COMPÉTENCES COUVERTES

- Maquetter une application
- Développer des composants d'accès aux données
- Développer la partie front-end d'une interface utilisateur web
- Développer la partie back-end d'une interface utilisateur web
- Concevoir une base de données
- Mettre en place une base de données
- Développer des composants dans le langage d'une base de données
- Concevoir une application
- Développer des composants métier
- Construire une application organisée en couches
- Développer une application mobile
- Préparer et exécuter les plans de tests d'une application



AUNAY VTT

# APPLICATION ENDURO

## 2- CONTEXTE DU PROJET

A black silhouette of a person riding a bicycle, positioned behind the text '2- CONTEXTE DU PROJET'.





AUNAY VTT

# APPLICATION ENDURO

## A- PRÉSENTATION DU CAPB

### 1- Qu'est ce que c'est ? De quoi s'agit-il ?

Le but statuaire de l'association est de développer et d'organiser des activités et manifestations culturelles, sportives ou de loisir principalement sur le territoire du Pré Bocage. L'association est assez ancienne. Elle a été créée en 1976 par un groupe d'habitants et de commerçants qui souhaitaient dynamiser les trois cantons Aunay sur Odon, Caumont l'Eventé et Villers Bocage. L'association s'est d'ailleurs longtemps appelée Comité d'Animation en Pré Bocage. Ces personnes avaient une imagination débordante. Ils ont organisé des dizaines d'événements variés comme des balltraps, des braderies commerciales, des soirées théâtre, des courses cyclistes, des repas dansants.

Actuellement l'association compte cinq activités. Un club de badminton « AC Bad » qui signifie Aunay Cahagnes Badminton, un club de VTT « Aunay VTT », un club de bridge, un club de football de table - activité plus connue sous le nom de babyfoot - « Aunay Football de Table » et le petit dernier un atelier d'Art Plastique « l'Atelier ».

Début 2020, l'association comptait 237 membres répartis ainsi : badminton 130, VTT 72, bridge 17, football de table 10 et arts plastiques 8. Sur ce total, 40% sont des jeunes de moins de 18 ans.

Cette association a fait émerger certains talents comme Damien Marie en 2003 qui remporte le titre de champion de France UFOLEP en VTT. Plus récemment des coureurs élites ont évolué au plus haut niveau national en FFC avec des résultats plus qu'honorables à ce niveau.

Et que dire des titres de champion du monde ou vice-champion du monde décrochés par des membres du babyfoot !

La crise sanitaire a eu des conséquences pour l'association, toutes les activités en intérieur sont à l'arrêt. Cela a un impact fort sur le nombre d'adhérent. Seul le VTT a pu poursuivre une vie de club mais malheureusement sans possibilité d'organisation et sans compétition.



AUNAY VTT

# APPLICATION ENDURO

## A- PRÉSENTATION DU CAPB

### 2- L'organisation au seins de l'association.

Les activités sont gérées chacune par une commission de 2 à 10 personnes. Chaque commission est quasiment autonome. Le conseil d'administration de l'association a plus un rôle de conseil de surveillance et est le garant de la bonne gestion de l'association dans son ensemble. Ce fonctionnement en multi activité permet aussi de mettre en commun des ressources ou du matériel.

Association Sports et Loisirs en Pre Bocage		
Jérôme PIOLINE	Président	Responsable section VTT
Julie LEBRETON	Vice-présidente	Section badminton
Gérard JUIN	Trésorier	Section badminton
Rémy PERNELLE	Secrétaire	Section VTT
Nicolas HERBINIERE	Membre	Section VTT
Jean-Christophe PORSON	Membre	Section badminton
Christophe CORDIER	Membre	Section football de table
Ourida PIOLINE	Membre	Responsable section badminton
Olivier ALLAIN	Membre	Section VTT
Alain HUET	Membre	Section badminton
Philippe SICOT	Membre	Responsable section bridge
Cathy LECORE	Invitée	Responsable section Art plastique

Détail de la commission VTT : Club AUNAY VTT		
Jérôme PIOLINE	Responsable de section	Entraîneur club
Rémy PERNELLE		Entraîneur club
Nicolas HERBINIERE		Entraîneur club
Maxime FEUGERES		Entraîneur club
Peter SIDLER		Entraîneur club
Tom HAVIN		Entraîneur club
Olivier ALLAIN		Animateur
Paul ROCHER	Expert X Country	Animateur
Tanguy MARIE		Animateur
Yannis CYSIQUE	Expert trial	Animateur
Jérémie MOREL	Expert enduro	



AUNAY VTT

# APPLICATION ENDURO

## B-LE BESOIN

### 1- Le cahier des charges

#### Objet

Le club Aunay VTT organise des épreuves de VTT Enduro depuis plusieurs années et souhaite disposer d'une application sur smartphone pour gérer et établir les classements de ce type d'épreuve. Jusqu'alors des fichiers Excell étaient utilisés mais ne donnaient pas satisfaction.

#### Définition

L'enduro est une des disciplines du VTT (Vélo Tout Terrain).

Les compétitions de VTT ENDURO sont des épreuves composées par une alternance de spéciales à profil descendant (sections chronométrées) et de liaisons (sections non chronométrées mais à réaliser dans un temps imparti). Le classement est effectué sur la base du cumul des temps mis par chaque coureur à parcourir l'ensemble des spéciales (chronométrage individuel). Une épreuve peut se dérouler sur une ou 2 journées.

#### Les Participants

Les participants s'inscrivent et règlent leurs droits d'inscription sur un site spécialisé. Une fois les inscriptions closes quelques jours avant l'épreuve, l'organisateur en extrait un fichier .xls qui comporte les champs suivants : Nom, Prénom, date de naissance, club, catégorie. L'organisateur ajoute ensuite un champ « Dossard » et affecte à chaque participant un numéro de dossard. Les numéros de dossard servent d'ordre de départ lors des différentes spéciales de l'épreuve. Les participants conservent le même numéro de dossard durant toute l'épreuve.

La liste des participants en .xls doit pouvoir être importée dans l'application avant le début de l'épreuve. Des corrections ou des ajouts de concurrents doivent pouvoir être réalisés directement dans l'application. Une fois figée la liste doit être disponible sur tous les smartphones du dispositif.

#### Les spéciales

Une épreuve peut comporter jusqu'à 20 spéciales sur deux jours. Elles sont numérotées de 1 à 20 s'il y en a 20. Les participants enchainent les spéciales dans l'ordre de la numérotation



AUNAY VTT

# APPLICATION ENDURO

## B-LE BESOIN

### 1- Le cahier des charges

En plus de leur numéro il est utile de leur donner à chaque spéciale un nom évocateur par exemple : « samedi descente du diable » ou « dimanche mass start »....

L'organisateur définit également pour chaque spéciale l'heure de départ du premier participant. L'application doit calculer l'heure de départ des participants suivants en fonction de l'intervalle préalablement défini. Exemple pour spéciale 1 : dossard N°1 – 13:00:00 puis dossard N°2 – 13:00:30 puis dossard N°3 – 13:01:00 ect...

Pour information le temps de parcours des spéciales est compris entre 1 et 5minutes.

#### Les liaisons

Bien que les liaisons ne soient pas chronométrées, elles sont préalablement reconnues par l'organisateur pour estimer un temps de parcours qui permet de définir l'heure du premier départ de chaque spéciale.

Dans le cas où un participant arrive en retard pour son départ, l'organisateur l'autorise à courir entre 2 concurrents mais son retard ne sera pas déduit de son chronométrage.

Pour information, les liaisons sont balisées et sécurisées afin que les participants ne s'égarent pas.

#### Dispositif de chronométrage – commissaires

Les équipes de commissaires sont composées de 4 personnes par spéciale : 2 commissaires au départ et 2 à l'arrivée. Chaque équipe est susceptible de se déplacer pour officier sur plusieurs spéciales.

Au départ, les commissaires s'assurent du bon ordre de départ des concurrents en vérifiant les numéros des dossards. Ils donnent le « start » à chaque participant à l'heure prévue. Pour cela ils ont à disposition une horloge ou bien une application qui gère un compte à rebours à l'intervalle prévu. Bien que non requise dans le développement à faire cette fonction intégrée à l'application serait un plus.



AUNAY VTT

# APPLICATION ENDURO

## B-LE BESOIN

### 1- Le cahier des charges

A l'arrivée, l'un des commissaires appelé aboyeur, est positionné quelques dizaines de mètres avant la ligne. Il annonce le numéro de dossard qui arrive. Le deuxième commissaire dispose sur son écran de smartphone de la liste des concurrents dans l'ordre des départs.

Il surveille la ligne et « sur le numéro annoncé lorsque le concurrent franchi la ligne. L'application enregistre l'heure d'arrivée. Il est possible qu'un concurrent se fasse doubler par un ou plusieurs autres coureurs ou pire encore qu'il n'arrive jamais (chute, panne...).

Dans ce cas, toutes les facilités doivent être prévues dans l'application pour que les commissaires puissent faire des corrections (inversion de concurrent, modification de l'heure d'arrivée...).

Il est souhaitable que l'application affiche instantanément le temps de chaque coureur par différence entre heure de départ et heure d'arrivée.

Il faut prendre en compte le fait que l'application doit pouvoir fonctionner hors ligne puisque certaines zones en pleine nature ne sont pas couvertes.

#### Les classements

A l'issue de l'épreuve une fonction de l'application doit permettre de compiler les données des différents smartphones utilisés sur les spéciales.

Le classement est établi en additionnant le temps de chacune des spéciales.

Le vainqueur étant celui qui aura mis le moins de temps au total. Cependant l'organisateur doit pouvoir sélectionner les spéciales à inclure dans le classement par exemple pour faire le classement d'une journée sur les 2 ou pour ne pas tenir compte d'une spéciale annulée...

L'organisateur a besoin du classement par catégories et du classement scratch (toutes les catégories).

Les classements doivent se présenter sous forme d'un tableau incluant le temps de chaque spéciale retenues, la somme des temps (voir ci-dessous) :



AUNAY VTT

# APPLICATION ENDURO

## B-LE BESOIN

### 1- Le cahier des charges

	Prénom	Nom	Catégorie	Club	Sp1	Sp2	Total
1	Jean	Cagny	Sénior	Néant	02:30:00	01:30:00	04:00:00
2	Pierre	LeGrand	Master	Néant	02:40:00	01:40:00	04:20:00
3	Alissia	Morel	Feminine	AunayVTT	02:42:00	01:42:00	04:24:00

Les classements doivent pouvoir être exportés en .pdf

Option : utilisation d'une cellule pour l'arrivée.

Afin d'être plus précis dans le chronométrage, l'organisateur souhaiterait remplacer le « clic » manuel par le signal tout ou rien (contact sec) d'une cellule photoélectrique ou laser. Il faudrait alors que ce signal soit récupéré dans le smartphone.



# B-L'ORGANISATION

Ce projet étant mis en place sur le temps libre des différents intervenants, la dead line était assez large. Le prochain enduro devrait avoir lieu en juillet 2022 donc le retard peut être toléré sur les tests en condition.

En cour de développement

### Dead-line



AUNAY VTT

# APPLICATION ENDURO

## B-L'ORGANISATION

### 2- Le client au coeur de la conception

Des réunions sont programmées une à deux fois toutes les semaines afin de faire un point sur l'avancée du projet et de répondre à certaines interrogations. Un drive a été mis en place afin de noter les interrogations qui se posent entre chaque réunion et pour y noter les réponses et les attentes du client face à chaque problématique posée.

En cas de problématique réellement bloquante pour l'avancée du projet il est toujours possible de contacter le client au téléphone ou par message afin d'avoir des précisions sur ses attentes.


Chaque étape de la conception front-end ainsi que chaque fonctionnalité sont validées par le client afin qu'en cas d'incompréhension les choses puissent être mises au clair le plus rapidement possible.





AUNAY VTT

APPLICATION  
ENDURO



# 3- CONCEPTION ET DEVELOPPEMENT DU PROJET



AUNAY VTT

# APPLICATION ENDURO

## A- SPÉCIFICATIONS TECHNIQUES

### 1- Les technologies utilisés

#### L'application mobile

##### Langage

- Java : Java est le langage Android le plus utilisé dans le développement mobile, de plus c'est un langage de programmation très réputé avec de vastes outils et bibliothèques open source. La communauté autour de ce langage est énorme ce qui implique de nombreuses ressources disponibles.
- XML : Ce langage de balisage est utilisé pour gérer l'affichage des contenus sur l'écran. Ce langage facilite le développement en permettant de séparer l'affichage des algorithmes.

##### Environnement de développement :

- Android Studio : Cet IDE est complètement dédié au développement Android. De plus il tourne sur tout les principaux systèmes d'exploitation. Il contient un éditeur de code, des outils de débogage et un émulateur.

##### Type de données particulière :

- UUID : UUID ( identifiant unique universel ) représente une valeur sur 128 bits. Ici elle permettra de définir l'identifiant pour chaque ligne de données de la table participation. Il est généré à chaque création d'un nouvel objet participation.
- TimeStamp : utilisé pour le traitement des dates des spéciales, ainsi que pour calculer les résultats de chaque participant.



AUNAY VTT

# APPLICATION ENDURO

## A- SPÉCIFICATIONS TECHNIQUES

### 1- Les technologies utilisés

#### L'API

Framework :

- **Symfony** : Symfony est un cadre MVC (Modèle Vue Contrôleur) développé en PHP. Symfony a l'avantage d'être livré avec des outils complémentaires qui permettent les tâches routinières et donc de se concentrer sur des fonctionnalités plus spécifiques. Les composants Symfony peuvent être utilisés de manière autonome dans d'autres frameworks ou dans des solutions PHP simples.
- **API Platform** : API Platform est un framework full stack puissant mais facile à utiliser dédié aux projets pilotés par API. Il permet de créer rapidement et simplement de puissantes API mais également sa documentation au format OpenAPI.



AUNAY VTT

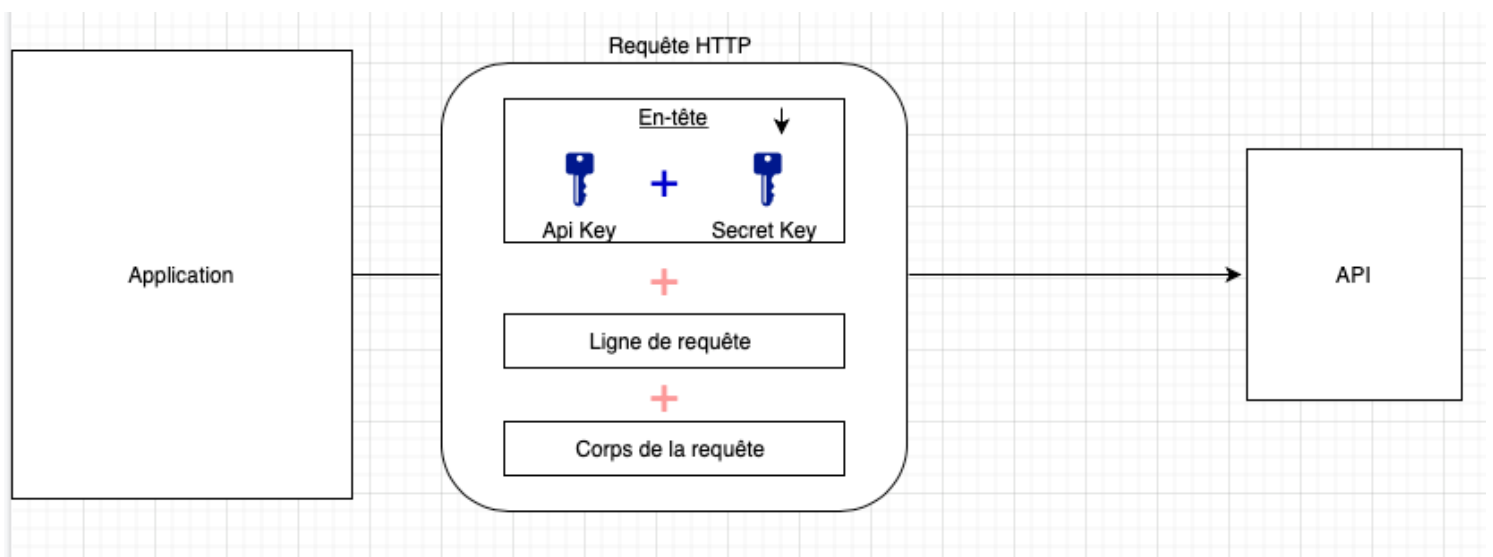
# APPLICATION ENDURO

## A- SPÉCIFICATIONS TECHNIQUES

### 2-La sécurité

La difficultés ici était qu'il fallait gérer le fait que les utilisateurs ne se connectent pas. Il faut donc faire en sorte que l'application elle même soit identifié auprès de l'API pour pouvoir se connecter à celle-ci. Pour cela on va utiliser l'authentification par clé API. A chaque appel à l'API, le client devra transmettre un couple Api Key/ secret key. La clef API sert à identifié le client, cette clef contient un identifiant permettant à l'API de reconnaître le responsable de la demande. Mais cette clef n'est pas secrète, alors on inclura dans chaque requête une clef secrète que seul le client et l'API connaîtront afin de garantir l'identité de l'émetteur.

Le client devra envoyer ces 2 clefs en basic auth ( Header Authorization Basic ). Auth utilise le protocole http et y ajoute une entête contenant justement nos deux clefs.



On ajoute à ça le protocole tls qui va permettre encore une fois à l'API d'identifier le client, notre application au certificat que ce protocole lui aura fournit.

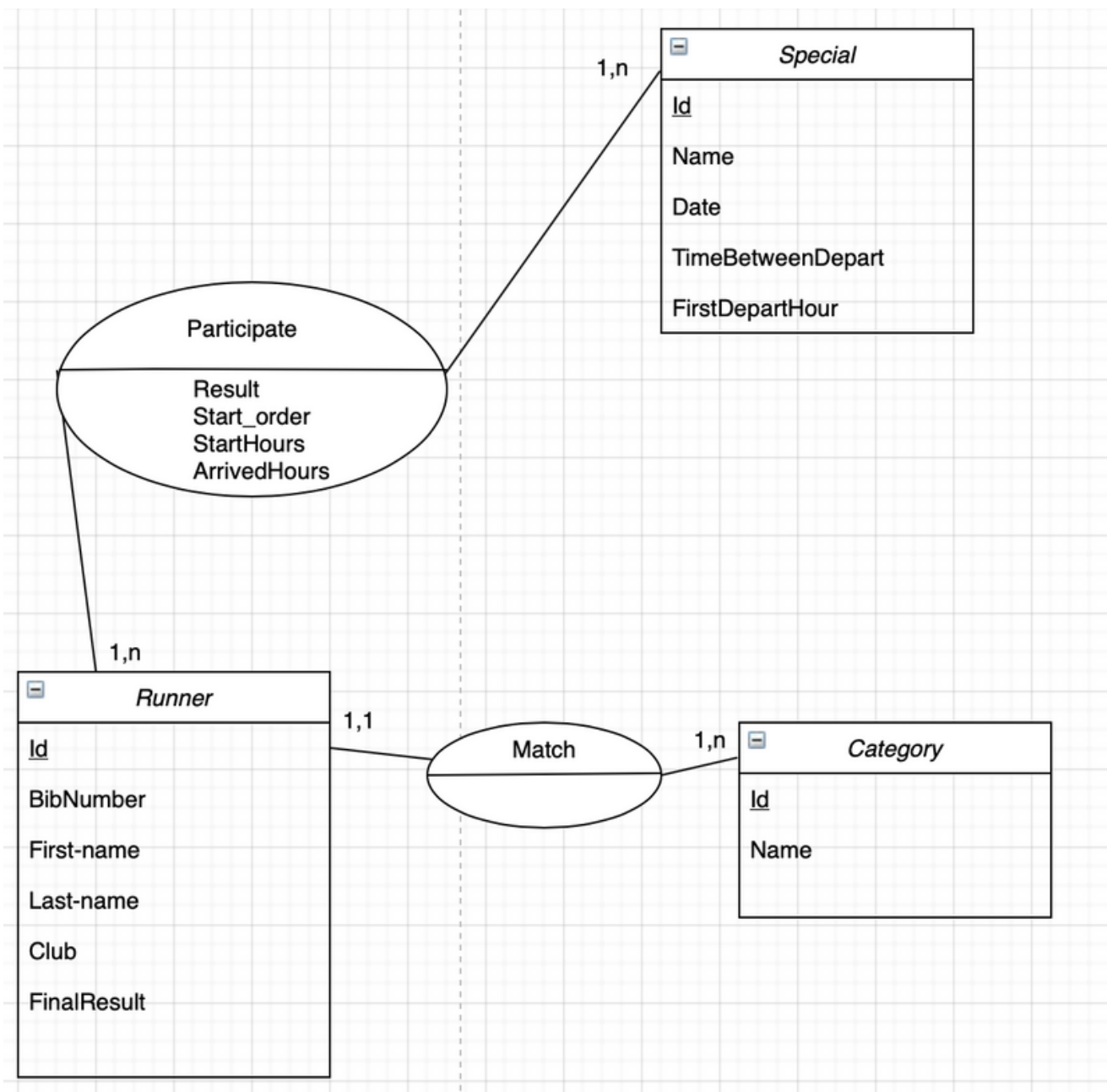


AUNAY VTT

# APPLICATION ENDURO

## B- BASE DE DONNÉES

### 1- MCD et relations





AUNAY VTT

# APPLICATION ENDURO

## B- BASE DE DONNÉES

### 1- MCD et relations

- Un coureur ne peut *correspondre* qu'à une seule catégorie
- Une catégorie peut *correspondre* à un ou plusieurs coureurs
- Un coureur peut *participer* à une ou plusieurs spéciales
- Une spéciale peut compter la *participation* d'un ou plusieurs coureurs



AUNAY VTT

# APPLICATION ENDURO

## B- BASE DE DONNÉES

### 2 - Dictionnaire de données

#### Category

Nom	Type	Longueur	Précision
Id	Integer auto_increment		Identifiant unique de la catégorie
Name	Text	255	Nom de la catégorie

#### Runner

Nom	Type	Longueur	Précision
Id	Integer auto_increment		Identifiant unique du coureur
<u>BibNumber</u>	Integer		Numéro de dossard du coureur
First-name	Text	255	Prénom du coureur
Last-name	Text	255	Nom du coureur
Club	Text	255	Club auquel appartient le coureur
<u>FinalResult</u>	Integer		Temps final réalisé par le coureur sur toute la course



AUNAY VTT

# APPLICATION ENDURO

## B- BASE DE DONNÉES

### 2 - Dictionnaire de données

#### Special

Nom	Type	Longueur	Précision
Id	Integer auto_increment		Identifiant unique du coureur
Nom	Text	255	Numéro de dossard du coureur
<u>TimeBetweenDepart</u>	Integer		Prénom du coureur
FirstDepartHour	Integer		Nom du coureur

#### Participation

Nom	Type	Longueur	Précision
Id	Integer auto_increment		Identifiant unique du coureur
Spécial	Integer		Numéro de dossard du coureur
Runner	Text	255	Prénom du coureur
Result	Text	255	Nom du coureur
<u>StartOrder</u>	Text	255	Club auquel appartient le coureur
DepartHour	Text	255	Heure de départ du coureur pour la spécial
<u>ArrivedHour</u>	Text	255	Heure d'arrivée du coureur pour la spécial





AUNAY VTT

# APPLICATION ENDURO

## B- BASE DE DONNÉES

### 3- Intégration à l'application

Sqlite est une base de données open source intégré dans chaque appareil mobile Android. C'est pour cela que l'on va l'utiliser afin de pouvoir stocker les données hors connexions.

Pour la création de la base de données une classe a été créé. Cette classe s'appelle DBOpenHelper et hérite de la classe SQLiteOpenHelper qui est une classe d'assistance pour la création de bases de données et pour la gestion de versions.

Au sein de cette classe DBOpenHelper une sous classe à été créé afin d'y stocker les constantes que l'on utilisera plus tard pour la requête de création de la base de données ( voir l'extrait de code suivant ) .



AUNAY VTT

# APPLICATION ENDURO

## B- BASE DE DONNÉES

### 3- Intégration à l'application

```
public static class Constants implements BaseColumns {

    /**
     * @goals This class aims to show the constant to use for the DB.DBOpenHelper
     */
    public static final String TAG = "SQLite";

    // The database name
    public static final String DATABASE_NAME = "enduro9.db";

    //The database version
    public static final int DATABASE_VERSION = 2;

    //The structure for the table 'category'
    public static final String TABLE_CATEGORIE = "category";
    public static final String COLUMN_CAT_CODE = "code";
    public static final String COLUMN_CAT_LIBELLE = "name";

    //The structure for the table 'runner'
    public static final String TABLE_PARTICIPANT = "runner";
    public static final String COLUMN_PAR_ID = "id";
    public static final String COLUMN_PAR_DOSSARD = "bibNumber";
    public static final String COLUMN_PAR_NOM = "lastName";
    public static final String COLUMN_PAR_PRENOM = "firstName";
    public static final String COLUMN_PAR_CLUB = "club ";
    public static final String COLUMN_PAR_RESULT = "finalResult ";
    //foreign key category
    public static final String COLUMN_PAR_CATEGORY = "category";
```



AUNAY VTT

# APPLICATION ENDURO

## B- BASE DE DONNÉES

### 3- Intégration à l'application

Une fois que toutes les constantes ont été créées on va créer les requêtes de création de chaque table et les mettre dans des variables. Dans ces requêtes on réutilisera donc les constantes définies précédemment. Dans l'exemple suivant on voit la construction des requêtes de créations des tables category et runner :

```
private static final String TABLE_CATEGORY_CREATE = "CREATE TABLE "  
    + Constants.TABLE_CATEGORIE + "(" +  
    Constants.COLUMN_CAT_CODE + " integer primary key autoincrement, " +  
    Constants.COLUMN_CAT_LIBELLE + " text not null); "  
  
private static final String TABLE_PARTICIPANT_CREATE = "CREATE TABLE "+  
    Constants.TABLE_PARTICIPANT + "(" +  
    Constants.COLUMN_PAR_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +  
    Constants.COLUMN_PAR_DOSSARD + " INTEGER NOT NULL , " +  
    Constants.COLUMN_PAR_NOM + " TEXT NOT NULL," +  
    Constants.COLUMN_PAR_PRENOM + " TEXT NOT NULL," +  
    Constants.COLUMN_PAR_CLUB + " TEXT NOT NULL, " +  
    Constants.COLUMN_PAR_CATEGORY + " INTEGER NOT NULL, " +  
    Constants.COLUMN_PAR_RESULT + " REAL , FOREIGN KEY (" +  
    Constants.COLUMN_PAR_CATEGORY + ") REFERENCES " + Constants.TABLE_CATEGORIE  
    + "(" + Constants.COLUMN_CAT_CODE + ");";
```

On se retrouve finalement avec 4 variables correspondant aux requêtes de création des tables, c'est à dire Category, Runner, Special et participation.



AUNAY VTT

# APPLICATION ENDURO

## B- BASE DE DONNÉES

### 3- Intégration à l'application

Maintenant on va créer le constructeur de `DBOpenHelper` et où l'on va appelé la fonction `super()` provenant de notre classe mère (`SQLiteOpenHelper`).

```
/**
 * @param context
 * @param name
 * @param factory
 * @param version
 */
public DBOpenHelper(Context context, String name, SQLiteDatabase.CursorFactory factory,
int version) {
    super(context, name, factory, version);
}
```

Il faut ensuite redéfinir les méthodes `onCreate` et `onUpdate` pour créer et mettre à jour la base de données.

Dans la classe `onCreate` on va dans l'ordre executer les requêtes concernant les tables `Category`, `Runner`, `Special` et enfin `Participation` afin que nous n'aillons aucun problème lors de la création des clefs étrangères.



AUNAY VTT

# APPLICATION ENDURO

## B- BASE DE DONNÉES

### 3- Intégration à l'application

```
@Override
public void onCreate(SQLiteDatabase db) {
    // Create the new database using the SQL string Database_create
    db.execSQL(TABLE_CATEGORY_CREATE);
    db.execSQL(TABLE_PARTICIPANT_CREATE);
    db.execSQL(TABLE_SPECIAL_CREATE);
    db.execSQL(TABLE_PARTICIPATION_CREATE);

    //Insert the categories using addCategory function
    this.addCategory(db, Constants.value);
}
```

Pour mettre à jour la base de données on supprime d'abord dans l'ordre les tables pour ensuite rappeler la fonction onCreate.

```
@SuppressWarnings("LongLogTag")
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    Log.w("DB.DBOpenHelper", "Mise à jour de la version " + oldVersion
        + " vers la version " + newVersion
        + ", les anciennes données seront détruites ");
    db.execSQL("DROP TABLE IF EXISTS " + Constants.TABLE_PARTICIPATION);
    db.execSQL("DROP TABLE IF EXISTS " + Constants.TABLE_SPECIAL);
    db.execSQL("DROP TABLE IF EXISTS " + Constants.TABLE_PARTICIPANT);
    db.execSQL("DROP TABLE IF EXISTS " + Constants.TABLE_CATEGORIE);

    // Create the new one
    onCreate(db);
}
```



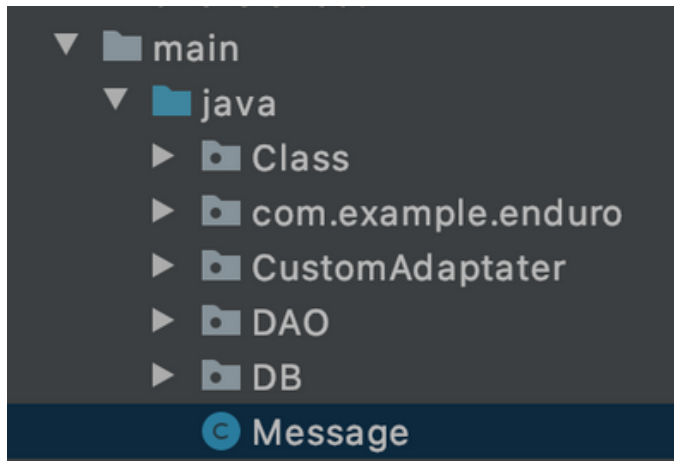
AUNAY VTT

# APPLICATION ENDURO

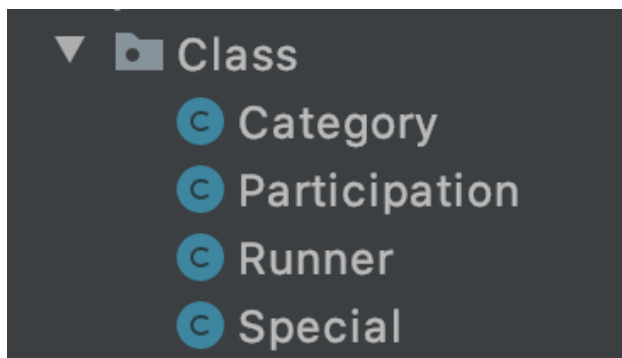
## C- L'APPLICATION MOBILE

### 1-L'arborescence

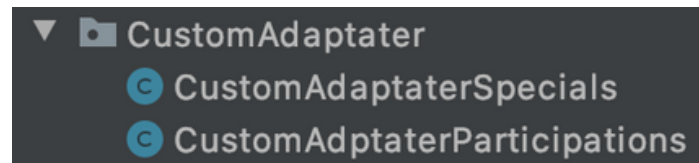
Les classes java sont classées de la manière suivante:



Le dossier Class contient toutes les classes d'objets.



CustomAdaptater lui contient les Adaptateurs de listView personnalisés :





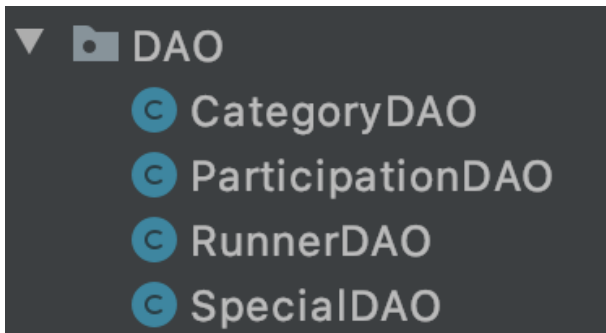
AUNAY VTT

# APPLICATION ENDURO

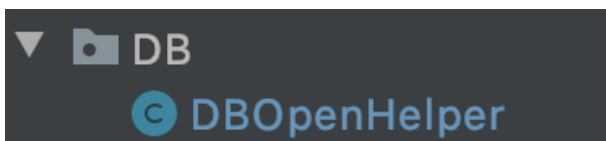
## C- L'APPLICATION MOBILE

### 1-L'arborescence

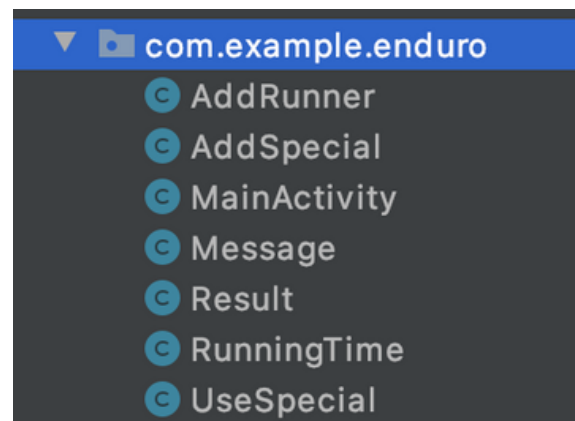
Dans le dossier DAO on retrouvera  
tout les fichiers DAO



Le dossier DB contient le  
fichier de création de base de  
donnée



Le dossier com.example.enduro  
lui contient les controllers pour  
chaque vue de l'application





AUNAY VTT

# APPLICATION ENDURO

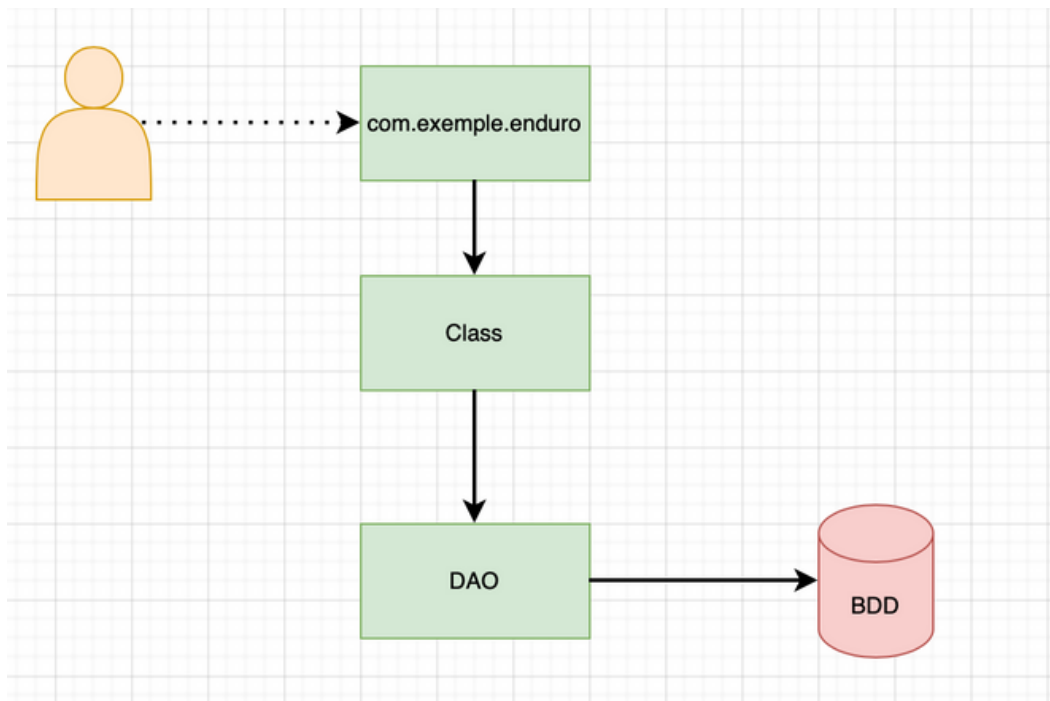
## C- L'APPLICATION MOBILE

### 1-L'arborescence

Avec cette arborescence on remarque vite la présence de 2 patterns : le pattern MVC et le pattern DAO.

Dans le pattern MVC le controleur gère l'interaction avec l'utilisateur et appellera le modèle pour effectuer des actions fonctionnelles ( comme la validation d'un formulaire ) et interagir avec la base de données.

Ici il a été choisi de séparer la persistance des données et les classes en elles même en ajoutant les classes DAO.



Les avantages sont que le traitement des données est séparé de la logique métier il est donc plus simple de mettre en place des tests unitaires. De même que si la structure de base de données change, il n'y aura pas à toucher à la logique métier.





AUNAY VTT

# APPLICATION ENDURO

## C- L'APPLICATION MOBILE

### 2- Logique metier et objet java

Dans cette application nous retrouverons 4 classes. La classe Category elle ne contient pas de logique particulière.

L'objet spécial est décrit par les attributs suivants :

- private int id
- private String name
- private String date
- private int timeBetweenDepart
- private String firstDepartHour

Les attributs id, name et date sont classiques. En revanche timeBetweenDepart correspond au temps entre chaque départ qu'on utilisera plus tard dans l'objet Participation. Pour l'attribut firstDepartHour c'est pareil, il correspond à l'heure du premier départ que l'on va convertir en timeStamp afin de s'en servir pour certains calculs.

L'objet Runner est décrit par les attributs suivants :

- private int id
- private int bibNumber
- private String firstName
- private String lastName
- private String club
- private int finalResult
- private Category category

Les attributs id, firstName, lastName et club sont eux aussi classiques et explicites par contre bibNumber correspond au numéro de dossard de chaque coureur ainsi qu'à leur ordre de départ. category correspond à l'objet Category correspondant à celle dont fait partie le coureur instancié. L'attribut finalResult correspond aux résultats final du coureur.



AUNAY VTT

# APPLICATION ENDURO

## C- L'APPLICATION MOBILE

### 2- Logique metier et objet java

Enfin l'objet participation correspond à la participation d'un coureur à une speciale et est décrit par les attributs suivants :

- private String id
- private Special special
- private Runner runner
- private int result
- private Timestamp startHours = new Timestamp(System.currentTimeMillis())
- private Timestamp arrivedHours = new Timestamp(System.currentTimeMillis())

L'attribut special correspond à la spéciale en cour pour cette participation et l'attribut runner correspond au coureur concerné par la participation.

Result lui correspond au temps mis par le coureur pour effectuer la spéciale.

On additionnera tout ces résultats pour calculer l'attribut finalResult de l'objet runner. L'attribut startHours lui est calculé en fonction du numéro de dossard et du temps entre chaque départ des class Runner et Special.

Si le numéro de dossard est le numéro 1 alors l'heure de départ correspond à l'attribut firstDepartHour de l'objet Special sinon on fait le calcul suivant :

$firstDepartHour + (bibnumber \times timeBetweenDepart)$ .

L'attribut arrivedHours, lui, correspond à l'heure exacte à laquelle ou l'agent spécifie que le coureur est arrivé.



AUNAY VTT

# APPLICATION ENDURO

## C- L'APPLICATION MOBILE

### 3- Bonnes pratiques et documentation

#### *Les bonnes pratiques*

- Chaque fonction est spécifiée et est commentée à l'intérieur même de la fonction ( voir le screen ci dessous ).
- Le code est indenté afin qu'il soit le plus lisible possible ( voir le screen ci dessous ).
- Le nom des fonctions et des variables est clair et compréhensible, le but étant qu'à la lecture du nom de la variable on sache à quoi elle sert. ( voir le screen ci dessous )

```
/**
 *
 * if the bibnumber is 1 so the start hours is the hour of the first depart because the depart order is about the bibnumber
 * else it's firstDepartHour + ( bibnumber * timeBetweenDepart )
 */
public void setStartHours(int bibNumber ) {

    //if the bibnumber is 1 so the start hours is the hour of the first depart because the depart order is about the bibnumber
    if(runner.getBibNumber()==1){

        String varTest = getSpecial().getFirstDepartHour();
        this.startHours = Timestamp.valueOf(getSpecial().getFirstDepartHour());

    }
    //if it's not the first so it's firstDepartHour + ( bibnumber * timeBetweenDepart )
    else{

        Timestamp firstDepartHour = Timestamp.valueOf(getSpecial().getFirstDepartHour());
        int timebetweenDepartHour = getSpecial().getTimeBetweenDepart();
        int toAdd = bibNumber*timebetweenDepartHour;
        this.startHours = new Timestamp(firstDepartHour.getTime() + (toAdd * 1000L));

    }
}
```



AUNAY VTT

# APPLICATION ENDURO

## C- L'APPLICATION MOBILE

### 3- Bonnes pratiques et documentation

#### Documentation

Pour la documentation l'outil javaDoc a été utilisé, il permet de générer des pages HTML contenant au minimum la liste des classes, la liste des méthodes et la liste des variables. Cela se présente sous cette forme :

All Classes

Packages

<unnamed package>  
Class  
com.example.enduro  
CustomAdaptater  
DAO  
DB

All Classes

AddRunner  
AddSpecial  
Category  
CategoryDAO  
CustomAdaptaterSpecials  
CustomAdptaterParticipations  
DBOpenHelper  
DBOpenHelper.Constants  
MainActivity  
Message  
Message  
Participation  
ParticipationDAO  
Result  
Runner  
RunnerDAO  
RunningTime  
Special  
SpecialDAO  
UseSpecial

OVERVIEW PACKAGE **CLASS** TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class

**Class Participation**

java.lang.Object  
Class.Participation

public class **Participation**  
extends java.lang.Object

Field Summary

Fields

Modifier and Type	Field and Description
private java.sql.Timestamp	<b>arrivedHours</b>
private java.lang.String	<b>id</b>
private int	<b>result</b>
private <b>Runner</b>	<b>runner</b>
private <b>Special</b>	<b>special</b>
private java.sql.Timestamp	<b>startHours</b>

Constructor Summary

Constructors

Constructor and Description
-----------------------------



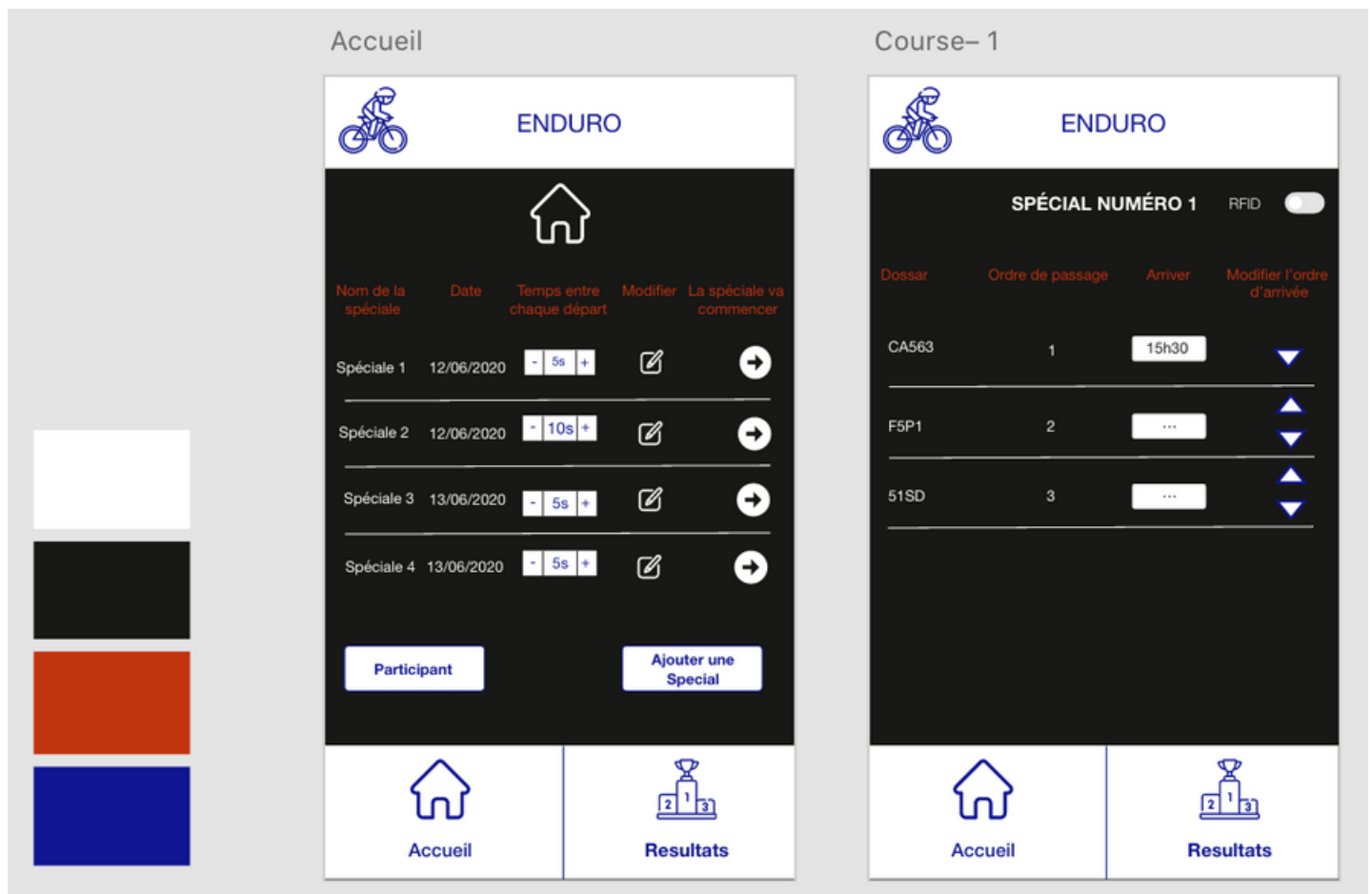
AUNAY VTT

# APPLICATION ENDURO

## C- L'APPLICATION MOBILE

### 4- Maquettes UX

Les maquettes suivantes ont été faites avec Adobe XD et ne correspondront pas aux visuels final de l'application. En effet elles ont été réalisées au tout début du projet et le cahier des charges a évolué depuis leur réalisation.





AUNAY VTT

# APPLICATION ENDURO

## C- L'APPLICATION MOBILE

### 4- Maquettes UX

Page d'accueil





AUNAY VTT

# APPLICATION ENDURO

## C- L'APPLICATION MOBILE

### 4- Maquettes UX

Page course en cours



ENDURO

**SPÉCIAL NUMÉRO 1** RFID ☐

Dossar	Ordre de passage	Arriver	Modifier l'ordre d'arrivée
CA563	1	15h30	▼
F5P1	2	...	▲▼
51SD	3	...	▲▼

**Accueil****Resultats**

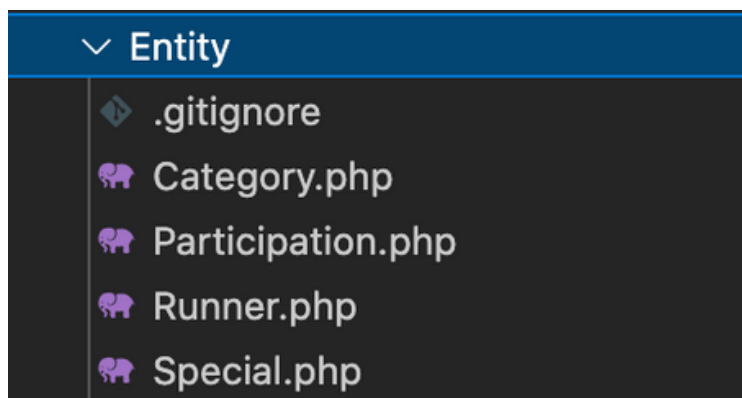
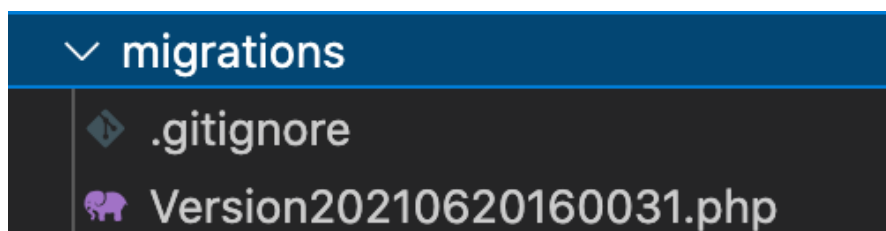


AUNAY VTT

# APPLICATION ENDURO

## D - L'API

### 1- L'arborescence et API plateforme



L'API étant développé avec symfony on retrouve l'arborescence classique de symfony avec ici un dossier migration qui contient les fichiers de migrations de création de base de données. On y trouve aussi un dossier src dans lequel se trouve un dossier Entity. C'est celui-ci qui nous intéressera le plus.

Dans le dossiers Entity chaque entité sera exposée comme ressource par l'API.

Pour se faire API plateforme à donc été utilisé.





AUNAY VTT

# APPLICATION ENDURO

## D - L'API

### 1- L'arborescence et API plateforme

```
/**
 * @ApiResource()
 * @ORM\Entity(repositoryClass=CategoryRepository::class)
 */
class Category
{
    ...
}
```

Pour exposer les entités comme ressource à l'API, l'annotation ( fournit par API plateforme ) `@ApiResource()` est utilisée. Toutes les opérations pour la ressource `Category` seront générées par API plateforme. Ces opérations sont fonctionnelles

API Plateforme utilise Swagger UI pour générer la documentation de l'API. La documentation prend cette forme :

Category			^
GET	/api/categories	Retrieves the collection of Category resources.	v
POST	/api/categories	Creates a Category resource.	v
GET	/api/categories/{id}	Retrieves a Category resource.	v
PUT	/api/categories/{id}	Replaces the Category resource.	v
DELETE	/api/categories/{id}	Removes the Category resource.	v
PATCH	/api/categories/{id}	Updates the Category resource.	v



AUNAY VTT

# APPLICATION ENDURO

## E- PHASE DE TEST

### 1- Test unitaire

Pour les tests unitaires sur l'API, la technologie php unit à été utilisée. C'est un framework open source de test qui est intégré à symfony. On va ici tester les fonctions indépendamment du reste de l'API.

Pour chaque entité un fichier de test à été créé afin de tester les getters comme dans l'exemple qui suit.

```
class RunnerTest extends ApiTestCase
{
    public function setUp() : void {
        parent::setUp();
        $this->runner = new Runner();
    }

    public function testGetBibNumber() : void{
        $value= 1;
        $response = $this->runner->setBibNumber($value);
        self::assertInstanceOf(Runner::class, $response);
    }

    public function testGetFirstName() : void{
        $value= "Jean";
        $response = $this->runner->setFirstName($value);
        self::assertInstanceOf(Runner::class, $response);
    }
    ....
}
```



AUNAY VTT

# APPLICATION ENDURO

## 4- CONCLUSION





AUNAY VTT

# APPLICATION ENDURO

## A- ERREUR ET RESOLUTIONS

- Soucis avec l'affichage de mon champ : firstDepartHour. Dans la requête de récupération getAllSpecial, l'heure s'affiche correctement mais dans mon adapter listView c'est l'heure actuelle qui s'affiche.

```
E/LOG: SELECT * FROM special
E/LOG: getAllSpecial heure du 1er depart : 2021-04-29 14:32:00.0
I/System.out: Exception :java.text.ParseException: Unparseable date: "2021-04-29 14:32:00.0"
E/LOG: getAllSpecial heure du 1er depart : 2021-04-29 14:36:00.0
I/System.out: Exception :java.text.ParseException: Unparseable date: "2021-04-29 14:36:00.0"
E/LOG: adapter : 2021-05-29 14:47:28.29
E/LOG: adapter : 2021-05-29 14:47:28.292
E/LOG: adapter : 2021-05-29 14:47:28.29
E/LOG: adapter : 2021-05-29 14:47:28.292
E/LOG: adapter : 2021-05-29 14:47:28.29
```

Solution : Utilisation d'une mauvaise méthode qui retournait la date actuelle. La date du premier détail était bien enregistrée en base mais une mauvaise valeur était affichée.

- La fonction getAllParticipationWithIdSpecial me renvoie beaucoup trop de résultats alors qu'il devrait être égal au nombre de coureur.

Solution : On ajoute les participations à chaque passage sur la vue runningTime. Il faut donc mettre une condition ou supprimer le contenu de cette table ( avec l'ID de la course correspondante ) dès que l'on revient sur une autre vue.



AUNAY VTT

# APPLICATION ENDURO

## B- ÉVOLUTIONS FUTURES ENVISAGÉES

- Développer l'application en swift pour IOS afin que les utilisateurs d'appareils IOS puissent participer en temps que commissaires.
- Développer une application web qui permettrait aux participants de consulter les résultats au fur et à mesure des épreuves.
- Développer la partie commissaire au départ avec une option connexion bluetooth et le top départ. automatique marqué par un trois "BIP" venant d'une enceinte.
- Automatisation de l'arrivée avec un capteur de signal tout ou rien (contact sec) d'une cellule photoélectrique ou laser. Il faudrait trouver dans ce cas la technologie hardware la plus adaptée aux contraintes des épreuves enduro.



AUNAY VTT

# APPLICATION ENDURO

## C- AVANT / APRÈS

Liste de toutes les spéciales :

Avant tout se faisait dans des tableaux excel. On rentrait les résultats pour chaque spéciale dans le même tableau comme ci dessous:

N°	Prénom	Nom	Sp 1	Sp 2	Sp 3	Sp 4	Sp 5	Sp6	Sp7
1	Basile	MARTINEAU	9:10:00	9:42:00	10:43:00	11:05:00	11:32:00	12:43:00	13:22:00
2	Jérémy	LEFEUVRE	9:10:30	9:42:30	10:43:30	11:05:30	11:32:30	12:43:30	13:22:00
3	Philippe	TESTUD	9:11:00	9:43:00	10:44:00	11:06:00	11:33:00	12:44:00	13:22:00
4	Milène	MOULIN	9:11:30	9:43:30	10:44:30	11:06:30	11:33:30	12:44:30	13:22:00
5	Bruno	LESTAS	9:12:00	9:44:00	10:45:00	11:07:00	11:34:00	12:45:00	13:24:00
6	Mathieu	LEGOUPIL	9:12:30	9:44:30	10:45:30	11:07:30	11:34:30	12:45:30	13:24:00
7	Olivier	BIENVENU	9:13:00	9:45:00	10:46:00	11:08:00	11:35:00	12:46:00	13:24:00
8	Yves	GUILLON	9:13:30	9:45:30	10:46:30	11:08:30	11:35:30	12:46:30	13:24:00
9	Gilles	FRIZAT	9:14:00	9:46:00	10:47:00	11:09:00	11:36:00	12:47:00	13:26:00
10	Gabriel	CARRO	9:14:30	9:46:30	10:47:30	11:09:30	11:36:30	12:47:30	13:26:00
11	Jules	LETOURNEUR	9:15:00	9:47:00	10:48:00	11:10:00	11:37:00	12:48:00	13:26:00
12	Fabien	BESSIN	9:15:30	9:47:30	10:48:30	11:10:30	11:37:30	12:48:30	13:26:00
13	Victor	BEAUDOIN	9:16:00	9:48:00	10:49:00	11:11:00	11:38:00	12:49:00	13:28:00
14	Fabien	LEBRETON	9:16:30	9:48:30	10:49:30	11:11:30	11:38:30	12:49:30	13:28:00
15	Valentin	LOUVEL	9:17:00	9:49:00	10:50:00	11:12:00	11:39:00	12:50:00	13:28:00
16	Dimitri	VAUTIER	9:17:30	9:49:30	10:50:30	11:12:30	11:39:30	12:50:30	13:28:00
17	Lewis	MAURIN	9:18:00	9:50:00	10:51:00	11:13:00	11:40:00	12:51:00	13:30:00

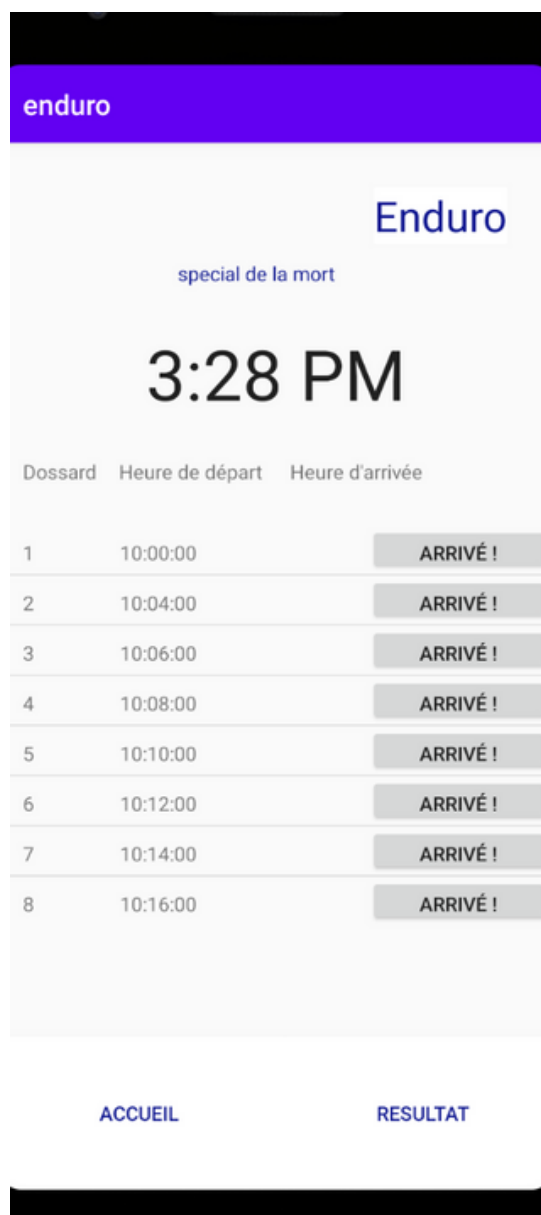


AUNAY VTT

# APPLICATION ENDURO

## C- AVANT / APRÈS

Dorénavant on trouve une vue avec la liste des spéciales, et lorsque l'agent clique sur la spéciale qu'il souhaite il retrouve l'ordre de départ et les informations concernant les participants pour cette spéciale.





AUNAY VTT

# APPLICATION ENDURO

## C- AVANT / APRÈS

La gestion des inscriptions continuera à se faire dans un tableau excel qui sera à importer dans la base de données. Il faut quand même gérer le cas d'une inscription tardive qui du coup se fera directement dans l'application via la vue suivante :

Nom	Prénom	Club	Catégorie	N° dossard
ALEM	Julien	Team Bikers 22	Junior	89
ANDRE	Nicolas	TEAM Armorique	Senior	72
AUBRY	Hugues	GROWPOW	Master 30	46
AUBRY	Paul	GROWPOW	Senior	47
AUMONT	Arnaud		Master 40	85
BEAUDOIN	Victor	EnDHuro 76	Master 30	13

The screenshot shows the 'enduro' application interface. At the top is a purple header with the word 'enduro' in white. Below the header, the title 'Enduro' is displayed in a large blue font. The form contains several input fields: 'Numero de dossard' (with a text input), 'Prenom' (with a text input), 'Club' (with a text input), 'Nom' (with a text input), and 'Categorie' (with a dropdown menu showing 'Senior'). Below these fields are two buttons: 'AJOUTER' and 'LISTE'. At the bottom of the form are two buttons: 'PARTICIPANT' and 'SPECIAL'. At the very bottom of the screen are two buttons: 'ACCUEIL' and 'RESULTAT'.





AUNAY VTT

# APPLICATION ENDURO

## D-TRAVAIL DE VEILLE

Pour ce projet il a fallut mettre en place une veille technologique poussée sur les outils utilisés et particulièrement concernant la sécurité. Pour cela les applications Feedly, twitter, YouTube et Spotify ont été utilisé mais aussi les sites lafermeduweb.net, développez.com, journalduhacker.net etc .

Une attention particulière a été portée sur la veille concernant la sécurité des API ( telle que les différentes méthodes d'authentification ) et sur les bonnes pratiques. Dans l'article suivant par exemple on retrouve une liste de bonnes pratiques concernant les API, illustrés par des exemples de code. <https://medium.com/pixelpoint/oh-man-look-at-your-api-22f330ab80d5> .

Des thèmes telle que la documentation, la gestion des erreurs ou l'utilité des statuts http y sont abordés.

Google drive et pocket sont les outils qui ont été utilisés afin d'enregistrer et de retrouver les articles, vidéos et podcasts.