



**Noms des élèves :**  
ERNEWEIN Cédric  
GAUTHIER Mathieu  
GUÉHO Damien  
GUYON Oriane  
KUOCH Thomas  
NDIAYE Serigne Fallou

**Commanditaire :**  
Équipe du Challenge 2016

**Tuteur(s) scientifique(s) :**  
MÜLLER Daniel

**Conseiller en communication :**  
MIRA-BONNARDEL Sylvie

**Conseiller en gestion de projet :**  
GIAUQUE Alexis

**Département d'accueil :**  
EEA

**Date du rapport :**  
10 Juin 2016

## **Titre complet du projet**

**Application Smartphone pour  
le Challenge Centrale Lyon**

# Résumé

Au coeur de tout objet de communication, les problématiques d'accès et de présentation de l'information sont primordiales. Il faut fournir rapidement et efficacement toute donnée jugée utile pour l'utilisateur.

Dans le cas du Challenge Centrale Lyon, 3000 participants arrivent sur le campus de l'école l'espace de deux jours pour participer à la compétition sportive des écoles d'ingénieurs la plus grande d'Europe. Les enjeux de communication sont bien réels et déterminants pour la réussite de l'événement puisqu'il s'agit de renseigner en direct 3000 étudiants répartis sur plus de 20 lieux sportifs différents.

L'objectif : créer une application smartphone téléchargeable par tout les participants.

Outre la création technique de cette application, l'enjeu est de donner accès efficacement aux informations nécessaires : lieux et horaires des matchs, infos utiles, numéros d'urgence, classement général, photos... Si le développement de ces fonctionnalités est bien d'ordre informatique, la qualité de l'expérience utilisateur qui doit être proposée est bien plus complexe à quantifier. Il s'agira de faire des choix pertinents et d'imposer de hauts standards d'excellence.

Mais le temps est court : tout se doit d'être opérationnel le week-end du 19 et 20 Mars 2016. Les jours sont comptés ! Il faudra être rapide et organisé.

Après une phase de réflexion et d'apprentissage des technologies utilisées, c'est toute la structure de l'application et de la base de données associée qui a été étudiée. Et c'est ce travail préliminaire qui a permis de coder une application entière en moins de 5 mois. Même si le travail de débogage a été intense, nous avons réussi à proposer l'application en téléchargement sur les deux plus grand magasins d'application du monde tout en étant prêt pour le Challenge.

Toujours dans un souci de qualité, le code final est fourni avec un manuel technique qui permet une prise en main facilitée pour quiconque désireux de poursuivre le projet.

# Abstract

At the heart of every communication device, access and display of information are key. Every data judged useful for the user must be given quickly and efficiently.

In the case of the Challenge Centrale Lyon, 3000 students gather on the school campus for two days, in order to participate to the biggest sport competition among engineer schools. Communication stakes are real and determinant for the success of the event, since it is a question of giving live information to 3000 students spread over 20 different sites.

The goal : creating a smartphone application downloadable by every participant.

Besides the application technical creation, the challenge is to give an efficient access to the information : place and time for the matches, useful information, emergency numbers, overall ranking, photos... If the development of these functionalities is about computer engineering, the user experience quality that must be proposed is way more complex to quantify. It will all be about making relevant decisions and imposing high standards of excellence.

But time flies : everything must be operational for the 19 and 20 March 2016 weekend. The clock is ticking! We will have to be quick and organised.

After a stage of thinking and learning about the technology used, the whole application structure and the associated data base that have been studied. And that preliminary study allowed us to develop a functional application in 5 months. Even if the debug time was under pressure, we achieved to propose the application in the two world largest online stores. And we were ready for the Challenge.

Since quality is our priority, the final code is provided with a technical manual that allows future work to be easier.

# Remerciements

Nous tenons avant tout à remercier M. Daniel MULLER pour nous avoir confié ce projet et aiguillé tout au long de l'année sur des aspects techniques. Ses remarques pertinentes et son expérience de développeur ont permis au projet de prendre la bonne direction.

Nous remercions également Mme Sylvie MIRA-BONNARDEL pour ses conseils judicieux et inédits sur l'aspect de la communication et de la présentation orale. Les séances étaient toujours vivantes et dynamiques, nous en tirerons les plus grands bénéfices.

Nos remerciements vont également à M. Alexis GIAUQUE pour ses précieux conseils en termes d'organisation du travail, de répartition des tâches et gestion du temps qui ont grandement participé à la réalisation à temps de ce projet.

Nous tenons tout particulièrement à adresser nos remerciements les plus sincères à M. Roland LAMIDIEU, ancien chef de projet du PE n°29 l'année dernière, qui a su rester disponible tout au long du projet pour nous orienter et nous rassurer sur nos choix. De plus, il nous a évité des erreurs qui auraient été une perte de temps.

Nous adressons également de chaleureux remerciements à Mme Lauranne BERTOSSI, responsable des relations avec notre projet d'étude au sein de l'équipe du Challenge, qui a su nous donner des conseils concernant le projet dans sa globalité ainsi que des remarques pertinentes concernant le design de l'application (et sa couleur).

Nous sommes très reconnaissants envers ÉCLAIR d'avoir accepté de nous prêter une partie de leurs serveurs pour héberger notre application.

Nous souhaitons aussi remercier tous les bêta-testeurs, Alexandre JOYEUX, Rémi RENAUDIE, Thomas PETIT, Thibaut VADCARD, Enoal LEPOITTEVIN et bien d'autres qui nous ont apporté des critiques utiles à l'amélioration de l'application. Nous remercions particulièrement Alexandre JOYEUX qui a participé au remplissage de la base de données quelques heures avant le Challenge.

Nous remercions également M. Raphaël MOULIN pour le temps qu'il nous a accordé au commencement du projet.

Enfin, nous remercions grandement toute l'équipe du Challenge 2016 sans qui ce projet n'aurait pas eu lieu d'être.

# Table des matières

<b>Résumé</b>	<b>2</b>
<b>Abstract</b>	<b>3</b>
<b>Remerciements</b>	<b>4</b>
<b>Glossaire</b>	<b>10</b>
<b>Introduction</b>	<b>14</b>
<b>1 Contexte du projet</b>	<b>15</b>
1.1 Besoins du Challenge . . . . .	15
1.2 Édition 2015 . . . . .	15
1.3 Bilan de l'édition 2015 : perspectives d'évolution . . . . .	18
1.4 Cahier des charges . . . . .	18
1.4.1 Rappel sur les demandes et contraintes du cahier des charges . . . .	18
1.4.2 Énoncé des besoins . . . . .	19
1.4.3 Synthèse du cahier des charges . . . . .	20
<b>2 État de l'art et bilan des connaissances : choix techniques</b>	<b>21</b>
2.1 Choix de la méthode : comment créer une application ? . . . . .	21
2.2 Choix du logiciel . . . . .	22
2.3 Choix de l'architecture : architecture MVC . . . . .	23
2.4 Architecture de l'application . . . . .	24
<b>3 Ouverture du projet et reflexion sur l'application</b>	<b>25</b>
3.1 Mise en place du frontend . . . . .	25
3.2 Mise en place du backend . . . . .	26
3.3 Vue globale de l'application . . . . .	27
3.4 Gestion de la base de données de l'application . . . . .	28
3.4.1 Architecture de la base de données . . . . .	30
3.4.2 Schéma relationnel de la base de données . . . . .	31
3.5 Travail sur l'expérience utilisateur (User Experience) . . . . .	32
3.6 Total des lignes de code . . . . .	32

<b>4</b>	<b>Une application mobile inédite : Challenge ECL</b>	<b>33</b>
4.1	Télécharger l'application . . . . .	33
4.2	Présentation de l'application . . . . .	34
4.2.1	Home . . . . .	34
4.2.2	Galerie . . . . .	35
4.2.3	Espace connexion . . . . .	35
4.2.4	Matches . . . . .	36
4.2.5	Ecoles . . . . .	40
4.2.6	Classement . . . . .	40
4.2.7	Informations . . . . .	41
<b>5</b>	<b>Dispositifs de communication</b>	<b>45</b>
5.1	La communication externe à l'École . . . . .	45
5.1.1	Facebook . . . . .	45
5.1.2	Vidéo . . . . .	45
5.2	La communication interne à l'École . . . . .	46
5.2.1	Communication pour les VP Tournois . . . . .	46
5.2.2	Communication avec l'équipe du Challenge . . . . .	47
5.2.3	Communication avec le PE 90 Challenge durable . . . . .	47
<b>6</b>	<b>Après le Challenge : résultats et analyses</b>	<b>48</b>
6.1	Téléchargements . . . . .	48
6.2	Retours des utilisateurs . . . . .	51
6.3	Analyse des retours . . . . .	52
6.3.1	Galerie photo . . . . .	52
6.3.2	La présentation des matchs . . . . .	52
6.3.3	L'entrée des sports par les VP Tournois . . . . .	53
<b>7</b>	<b>Bilan et perspectives futures</b>	<b>55</b>
7.1	Bilan de notre travail . . . . .	55
7.2	Comparaison . . . . .	56
7.3	Qu'en est-il pour la suite ? . . . . .	58
	<b>Conclusion</b>	<b>59</b>
<b>A</b>	<b>Communication pour les VP Tournois</b>	<b>60</b>
<b>B</b>	<b>Questionnaire de satisfaction pour les utilisateurs de l'application</b>	<b>62</b>
<b>C</b>	<b>Liste des News</b>	<b>65</b>
C.1	Vendredi . . . . .	65
C.2	Samedi . . . . .	65
C.3	Dimanche . . . . .	66
<b>D</b>	<b>Coûts</b>	<b>68</b>
D.1	Les coûts financiers . . . . .	68
D.2	Les coûts en temps . . . . .	68

<b>E</b>	<b>Présentation de l'équipe</b>	<b>70</b>
E.1	Membres de l'équipe . . . . .	70
E.2	Répartition des tâches . . . . .	70
<b>F</b>	<b>Diagrammes de Gant</b>	<b>72</b>
<b>G</b>	<b>Manuel technique</b>	<b>74</b>
G.1	Développement du Frontend . . . . .	74
G.1.1	Installation de Ionic . . . . .	74
G.1.2	Structure du dossier de travail . . . . .	74
G.1.3	Pré-visualiser l'application sur l'ordinateur . . . . .	76
G.1.4	Codage des différentes pages de l'application . . . . .	77
G.1.5	Le CSS . . . . .	93
G.2	Déploiement de l'application . . . . .	93
G.2.1	Compilation sous Android . . . . .	93
G.2.2	Compilation sous iOS . . . . .	95
G.3	Description du Backend . . . . .	95
G.3.1	Description de l'API . . . . .	95
G.3.2	Requêtes traitées par l'API . . . . .	96
G.3.3	Le coeur de l'API : index.php . . . . .	96
G.3.4	Le dossier models . . . . .	99
G.4	Conclusion . . . . .	100

# Table des figures

1	iOS 9 et Swift . . . . .	11
2	Android et Java . . . . .	11
1.1	Menu et Page d'accueil de l'application de 2015 . . . . .	16
1.2	Programmes du week-end et Plans . . . . .	16
1.3	Page Matches avec sports collectifs (phases de poules et phases finales) et sports individuels . . . . .	17
1.4	FAQ et numéros d'urgence . . . . .	17
2.1	Ionic . . . . .	22
2.2	Modèle MVC . . . . .	23
2.3	Architecture globale . . . . .	24
3.1	Fichiers de l'application . . . . .	25
3.2	Fichiers de l'API . . . . .	26
3.3	Modèle de l'application . . . . .	27
3.4	Page d'accueil du site internet permettant la gestion de la base de données de l'application . . . . .	28
3.5	Architecture globale finale . . . . .	29
4.1	Challenge ECL sur l'AppStore et le PlayStore . . . . .	33
4.2	Les <i>tabs</i> de l'application . . . . .	34
4.3	Page d'accueil de l'application . . . . .	34
4.4	Galerie photos; la dernière image représente une image zoomée . . . . .	35
4.5	Espace connexion . . . . .	36
4.6	Page Matches : sélection d'un sport et d'un genre . . . . .	36
4.7	Page Matches : recherche du tournoi de football féminin . . . . .	37
4.8	Page Matches : recherche du tournoi de judo masculin . . . . .	37
4.9	Page Matches : recherche du tournoi d'ultimate (tournoi mixte) . . . . .	38
4.10	Football féminin : phases de poules en bas et phases finales en haut . . . . .	39
4.11	Interface de la page Matches pour un VP Tournoi connecté (en Handball ici) . . . . .	39
4.12	Page Ecoles . . . . .	40
4.13	Page Classement . . . . .	41
4.14	Page Infos . . . . .	41
4.15	Section développement durable de la page Infos . . . . .	42
4.16	Ici, programme du Dimanche . . . . .	42
4.17	Plan du campus (affichage normal puis zoomé) . . . . .	43
4.18	La section urgences avec des numéros utiles . . . . .	43



4.19	FAQ . . . . .	44
5.1	Page Facebook . . . . .	45
5.2	Quelques captures de la vidéo proposée . . . . .	46
5.3	Logo du Challenge durable . . . . .	47
6.1	Graphique montrant le nombre de téléchargements de l'application (en bleu pour iOS et en vert pour Android) . . . . .	48
6.2	Graphique montrant la répartition des téléchargements de l'application (en bleu pour iOS et en vert pour Android) . . . . .	49
6.3	Graphique montrant la répartition des téléchargements de l'application selon l'appareil utilisé (en bleu pour iOS et en vert pour Android) . . . . .	49
6.4	Graphique montrant la répartition des téléchargements de l'application selon le territoire (en bleu pour iOS et en vert pour Android) . . . . .	50
6.5	Affichage du tournoi de volley masculin avec phases finales et phases de poule	53
6.6	Temps de remplissage moyen des résultats des matchs dans l'application par les VP Tournois . . . . .	53
A.1	Première page de la documentation distribuée aux VP Tournois . . . . .	60
A.2	Deuxième page page de la documentation distribuée aux VP Tournois . . .	61
B.1	Première partie du questionnaire . . . . .	62
B.2	Deuxième partie du questionnaire . . . . .	63
B.3	Troisième partie du questionnaire . . . . .	64
F.1	Organisation de la première période . . . . .	72
F.2	Organisation de la deuxième période . . . . .	73
G.1	Dossier de travail de l'application . . . . .	75
G.2	Pré-visualisation de l'application mobile sur ordinateur - Page Matches . . .	76
G.3	Barre de navigation de l'application . . . . .	77
G.4	Page Informations . . . . .	78
G.5	Page Classement . . . . .	80
G.6	Page Home . . . . .	82
G.7	Page Matches . . . . .	88
G.8	Page Ecoles . . . . .	90
G.9	Recherche des matchs de l'Ecole Centrale de Lyon . . . . .	92
G.10	Fichiers de l'API . . . . .	96
G.11	Liste des requêtes traitées par l'API . . . . .	97

# Liste des tableaux

1.1	Synthèse du cahier des charges . . . . .	20
3.1	Base de données <i>challenge2016</i> . . . . .	30
3.2	Schéma relationnel de la base de données . . . . .	31
4.1	Méthodologie d’affichage des sport . . . . .	38
6.1	Récapitulatif des téléchargements . . . . .	50
7.1	Comparaison de l’application 2016 Challenge ECL avec la web-app de 2015	57
D.1	Récapitulatif des dépenses . . . . .	68

# Glossaire

## iOS

C'est le système d'exploitation qui équipe tous les appareils mobiles d'Apple. Tous les iPhones fonctionnent sous iOS. Les iPhones les plus récents sont équipés d'iOS 9. Le langage de développement utilisé est Swift.

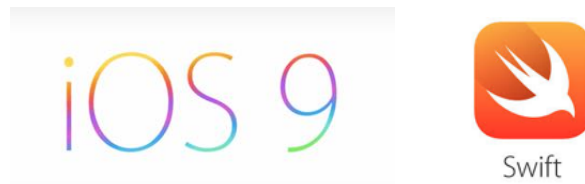


FIGURE 1 – iOS 9 et Swift

## Android

C'est le système d'exploitation qui équipe tous les appareils mobiles sous contrat avec Google. Samsung, HTC, Sony, Wiko, Huawei, et bien d'autres marques utilisent Android. Les mobiles les plus récents sont équipés d'Android 6. Le langage de développement utilisé est Java.



FIGURE 2 – Android et Java

## Application hybride

C'est une application ni purement native (donc codée en Swift pour iOS ou Java pour Android), ni purement basée sur du développement Web (HTML, CSS, PHP, Javascript...).

## Frontend

Le Frontend d'une application se compose des éléments que l'on voit à l'écran et avec lesquels on peut interagir. Le Frontend représente tout le design de l'application ainsi que son développement.

## **Backend**

Le Backend d'une application est sa partie immergée. Invisible pour les utilisateurs, elle représente la base de l'application avec notamment les communications avec le serveur et la base de données.

## **Framework**

Un framework est, comme son nom l'indique en anglais, un « cadre de travail ». L'objectif d'un framework est généralement de simplifier le travail des développeurs informatiques (codeurs), en leur offrant une architecture « prête à l'emploi » et qui leur permette de ne pas repartir de zéro à chaque nouveau projet. [1]

## **Newsfeed**

Méthode qui permet d'afficher sur la page personnelle d'un utilisateur les informations concernant les actions de son réseau social. [2]

## **API**

API est l'abréviation de Application Programming Interface (Interface Applicative de Programmation). Les API sont un moyen efficace de faire communiquer entre elles deux applications. Pour schématiser, on peut dire qu'une API crée un langage commun pour dialoguer et échanger des données entre une application A et une application B, qui pourtant ne sont pas programmées dans le même langage ni physiquement au même endroit. [3]

## **API Restful (ou REST)**

REST (Representational State Transfer) ou Restful est un style d'architecture permettant de construire des applications (Web, Intranet, Web Service). Il s'agit d'un ensemble de conventions et de bonnes pratiques à respecter et non d'une technologie à part entière. L'architecture REST utilise les spécifications originelles du protocole HTTP, plutôt que de réinventer une surcouche. [4]

## **Système de Gestion de Base de Données (ou SGBD)**

Un Système de Gestion de Base de Données est un logiciel qui permet de stocker des informations dans une base de données. Un tel système permet de lire, écrire, modifier, trier, transformer ou même imprimer les données qui sont contenus dans la base de données. [5] Nous utiliserons MySQL.

## **VP Tournoi**

Un VP Tournoi est un étudiant de l'École Centrale de Lyon qui participe à l'organisation du tournoi (programmation, durées, horaires, règles...) d'un sport spécifique. Il sera chargé d'entrer les résultats dans l'application.

## **ÉCLAIR**

C'est l'association de l'École Centrale de Lyon : Amis de l'Informatique et des Réseaux. ÉCLAIR possède des serveurs qui seront utilisés pour l'hébergement de la base de données.

### **Dessin vectoriel**

Répandu dans le monde professionnel et particulièrement dans le monde du graphisme, le dessin vectoriel est souvent utilisé pour améliorer le visuel de divers projets, dont les créations pour des sites web. Un dessin vectoriel est constitué à la base de formes simples comme des points, des lignes, des arcs, des cercles, des rectangles, etc. Ces bases peuvent subir des transformations simples pour former des formes plus complexes. [6]

### **Expérience utilisateur**

Le terme « Expérience Utilisateur » a été diffusé dans les années 90 par Donald NORMAN (auteur du célèbre "Design of everyday things" et aussi de la méthode de design centrée sur l'utilisateur) et, d'après la norme ISO 9241-210, correspond « aux réponses et aux perceptions d'une personne qui résultent de l'usage ou de l'anticipation de l'usage d'un produit, d'un service ou d'un système ». [7]

# Introduction

Notre projet d'étude n °29 " Application pour le Challenge", est un maillon d'un projet bien plus vaste : mener à bien le Challenge 2016. Il s'agit de rassembler le temps d'un week-end - les 19 et 20 Mars 2016 -, sur le campus de l'Ecole Centrale de Lyon, 3000 étudiants venus de toute la France et d'Europe pour une compétition sportive inter-écoles, agrémentée d'une soirée mémorable le samedi soir. Amener, nourrir, loger, répartir et occuper autant de personnes représente un défi de taille.

Dans ce contexte, notre projet d'étude a eu pour mission d'élaborer l'application smartphone de l'événement. Ses deux finalités principales sont de donner un accès permanent aux informations sportives du week-end (programmation, résultats...) et de faire découvrir l'événement aux étudiants qui ne connaissaient pas encore le Challenge.

En 2015, une application web avait été développée pour le Challenge. Cette application était satisfaisante puisqu'elle fournissait les informations demandées mais manquait cruellement d'ergonomie. La structure de fond avait été travaillée et réfléchi mais le design général pouvait être amélioré. De plus, le code n'était pas assez "propre" pour être réutilisé.

Nous sommes donc partis avec une feuille blanche (ou plutôt un écran vide) mais néanmoins avec un cahier des charges assez précis. Nous avons récupéré et analysé ce cahier des charges et après une phase d'analyse poussée où nous avons construit la structure entière du projet sur papier, nous nous sommes mis au travail.

5 mois de codage alternant du développement frontend et backend, des réflexions de design et d'ergonomie ainsi que des questionnements sur l'aspect de l'information proposée ont été nécessaires pour produire une application inédite : Challenge ECL. Nous vous présentons dans ce rapport tout notre travail de développement ainsi que notre travail de communication et notre travail d'analyse après le Challenge.

# Partie 1

## Contexte du projet

### 1.1 Besoins du Challenge

Le Challenge Centrale Lyon est un tournoi sportif de référence dans le monde étudiant. Si les élèves de plus de 34 grandes écoles d'ingénieurs françaises et européennes se déplacent chaque année à Lyon, c'est bien parce que la compétition sportive est remarquable.

Chaque année, le Challenge se renouvelle pour proposer aux participants toujours plus de qualités dans les sports, les activités, les soirées...

Il semble alors nécessaire que le Challenge utilise tous les moyens qu'il peut avoir à disposition, notamment les moyens numériques d'aujourd'hui. Sites internet et réseaux sociaux le sont déjà, mais pour renforcer sa présence digitale, une application mobile « Challenge » semble inévitable.

L'équipe de 2015 a alors commandé une application pour leur 33<sup>ème</sup> édition.

### 1.2 Édition 2015

En 2015, le Challenge Centrale Lyon disposait pour la première fois d'une web-app, soit une application web. A partir de n'importe quel mobile, il fallait ouvrir un navigateur (Chrome, Safari...) et se rendre à l'adresse `app.challenge-grandes-ecoles.fr` et l'on avait accès :

- à des news ;
- à la programmation des matchs ;
- aux informations concernant les navettes ;
- au programme du week-end ;
- aux plans ;
- à une FAQ ;
- aux informations concernant les urgences ;
- aux différents partenaires du Challenge.

Voici quelques aperçus de l'application (Figure 1.1) :

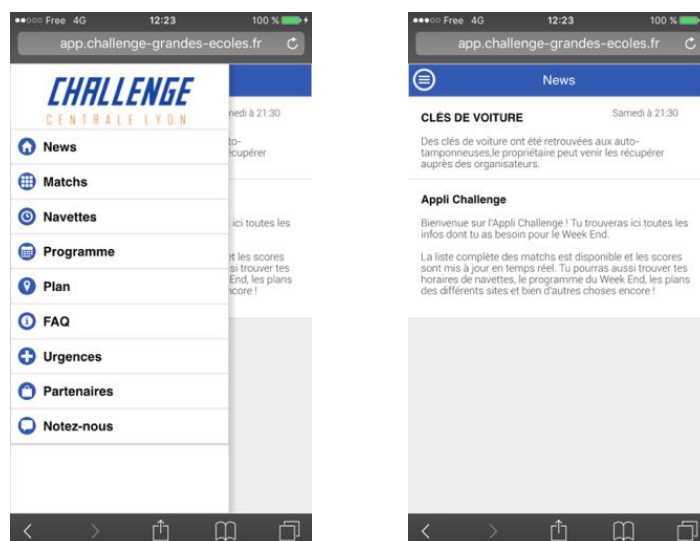


FIGURE 1.1 – Menu et Page d'accueil de l'application de 2015

La page *News* permet de donner en temps réel des informations aux participants : horaires du RU, deadlines pour acheter des billets, début de certains événements... ; soit tous les messages que l'équipe du Challenge veut faire passer aux participants.

Les participants avaient aussi accès au programme du week-end ainsi qu'aux plans des différents sites sportifs (Figure 1.2) :

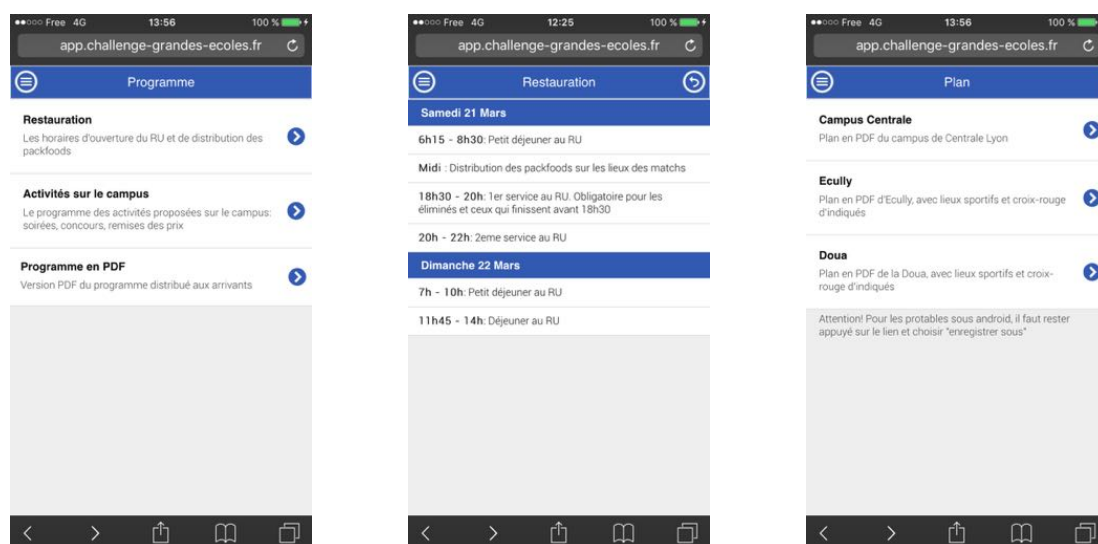


FIGURE 1.2 – Programmes du week-end et Plans

Tandis que la page *Programme* permettait aux différents participants de regarder à l'avance ce qu'ils voulaient faire du week-end (quel match regarder, à quelles activités participer), la page *Plan* leur permettait de se repérer sur le campus et sur les autres sites sportifs.



Peut-être l'un des endroits de la web-app les plus important, la page *Matches* (Figure 1.3) :

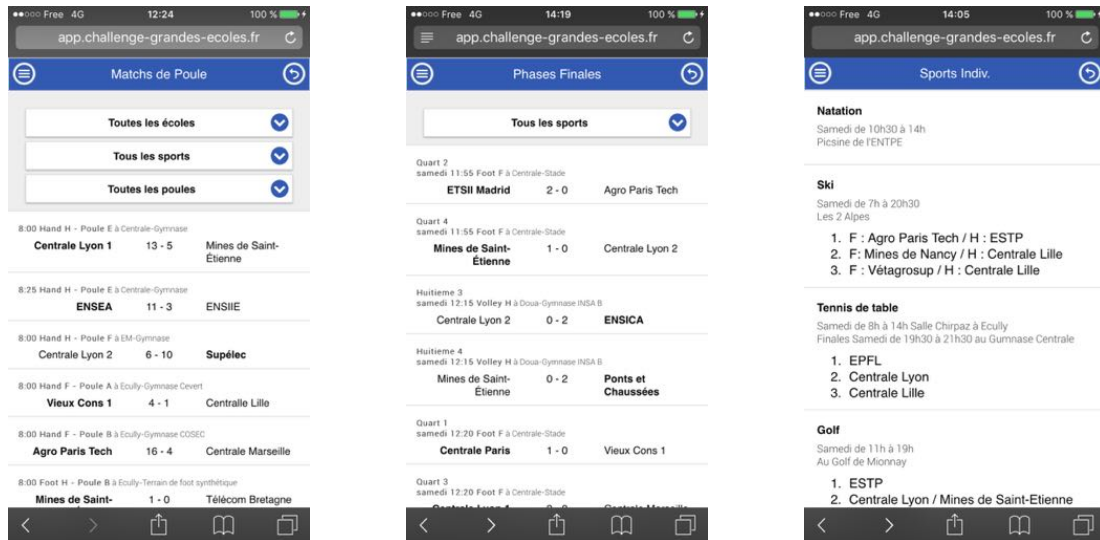


FIGURE 1.3 – Page Matches avec sports collectifs (phases de poules et phases finales) et sports individuels

Ainsi, tous les participants étaient informés des résultats sportifs de toutes les équipes dans tous les sports. Des menus déroulants permettaient de faire des recherches filtrées concernant les sports, les écoles et/ou les poules.

Enfin, les pages *FAQ* et *Urgences* permettaient d'avoir des informations pratiques concernant les sports, la communication, les soirées, la logistique... et d'obtenir des numéros de téléphones en cas de problème grave (Figure 1.4) :

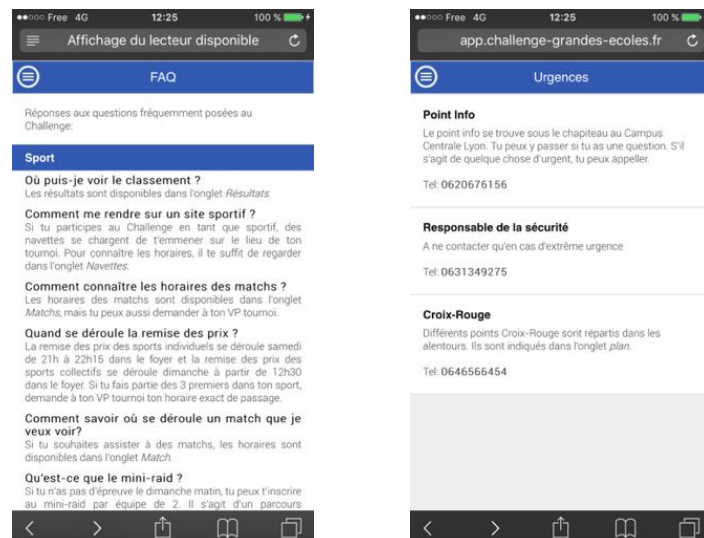


FIGURE 1.4 – FAQ et numéros d'urgence

## 1.3 Bilan de l'édition 2015 : perspectives d'évolution

Ce sont ici les réalisateurs de la web-app qui s'expriment.

*« L'un des principaux défauts qui a été formulé est que les résultats des matchs n'étaient pas assez mis à jour. Afin de contrer ce problème nous avons pensé à un système de serveur SMS : le VP Tournoi envoie un SMS avec le nom des écoles et le score. Ces données seront ensuite rentrées automatiquement dans l'application.*

*Nous avons pensé donner des codes d'accès pour l'interface staff aux VP Tournois, cependant cette solution n'est pas viable car cela implique qu'ils doivent se connecter à internet. De plus, nous avons eu vent du problème récurrent du manque de batterie, le fait de se connecter à internet ne va faire qu'accroître ce problème. »*

*« Le prochain PE pourrait se lancer dans la réalisation d'une « vrai Application smartphone ». Nous n'avons pas pu le faire cette année du fait que nous partions de zéro. Cependant le prochain PE possédera une base et pourrait donc se lancer dans un tel projet. A la vue de l'expérience du Challenge, nous pensons qu'une vraie application pourrait accroître encore l'audience. De plus l'avantage principal d'une application est que celle-ci permet l'envoi de notifications si l'utilisateur le permet. Ainsi, l'utilisateur recevrait directement en notification les news, et les scores des matchs qui l'intéressent. »*

*« Aussi, un certains nombres de pages sont statiques. Ceci présente l'inconvénient qu'à chaque nouvelle édition il faut se replonger dans le code de l'application pour pouvoir modifier les données. Rendre les pages dynamiques serait un avantage notable puisque l'application serait ensuite « autonome ». Elle pourrait ensuite être reprise chaque année sans que nécessairement quelqu'un maîtrise l'informatique. Nous avons déjà commencé à rendre des pages dynamique. »*

A la fin de l'année scolaire 2015, l'ancienne équipe a donc établi un cahier des charges pour une nouvelle application. C'est donc ce cahier des charges qui servira de commencement pour le développement de notre application.

## 1.4 Cahier des charges

### 1.4.1 Rappel sur les demandes et contraintes du cahier des charges

#### Les demandes

L'objectif principal est de permettre aux participants du Challenge Centrale Lyon d'avoir plus facilement accès aux informations qui leur sont utiles et ce à n'importe quel endroit du campus ou ailleurs et leur procurer des informations en temps réel sur les événements et les différents sports qui se déroulent.

L'application doit au minimum contenir les pages suivantes :

- Flux d'actualités/ FAQ ;
- Résultats ;
- Programme du week-end ;
- Horaires navettes - Plans ;
- Contact ;
- Partenaires.

## **Les contraintes**

L'application doit être une web-app adaptée aux smartphones suivants : iPhones et Android. L'application doit pouvoir supporter l'utilisation simultanée d'un grand nombre d'utilisateurs.

### **1.4.2 Énoncé des besoins**

Nous avons retenu 3 besoins fondamentaux :

- la simplicité d'utilisation de l'application ;
- la diffusion d'informations utiles pour le participant ;
- la possibilité d'actualiser l'application.

Nous envisageons aussi des besoins secondaires comme l'existence de l'application en version anglaise, une personnalisation de l'application adaptée au statut du participant (ex : membre de l'équipe de handball de l'ESTP, membre du staff...), une adaptation de l'application à d'autres formats (iPad), un onglet photos, un onglet « notez-nous » pour nous permettre d'avoir un retour sur la performance de l'application. Nous considérons actuellement ces besoins comme optionnels et non prioritaires.

## **Les besoins de simplicité d'utilisation**

L'application doit être utilisable par n'importe quel sportif muni d'un smartphone Android ou iPhone. Il faut que l'application soit simple d'installation. Il faut que l'utilisateur n'ait pas de problème à se connecter, à ouvrir l'application.

L'application a pour but d'informer les sportifs. Il est donc primordial que les sportifs trouvent facilement les informations qui les concernent.

Pour être simple d'utilisation, l'application doit être ergonomique et esthétique.

La simplicité de l'utilisation de l'application vaut aussi pour les membres de l'équipe du Challenge qui auront le besoin d'entrer des données sur le serveur, avant, pendant et après l'événement, et à une fréquence à déterminer.

## **Les besoins d'actualisation de l'information**

L'application doit pouvoir être mise à jour facilement par le staff. L'application doit être suffisamment dynamique et interactive. Les informations qu'elle fournit doivent pouvoir être actualisées à tout moment, facilement et rapidement. La rapidité de l'actualisation est un critère de réussite du projet.

## La diffusion d'informations utiles pour le participant

Ce besoin est double. Il s'agit d'apporter un moyen de communication adapté au participant pour une bonne diffusion et de bien définir ce qui caractérise une information utile pour le participant.

## Les besoins ne faisant pas partie du projet

### *Un réseau social*

L'application ne fournit pas la possibilité aux utilisateurs de communiquer entre eux, de partager des photos.

### *L'utilisation de Google Maps*

L'application ne géo-localise pas l'utilisateur sur le campus. Pour les sites extérieurs, l'utilisateur utilisera des liens qui le renvoient sur le Google Maps. Elle n'est pas prévue dans l'application Challenge.

## 1.4.3 Synthèse du cahier des charges

Fonctions principales	Sous-fonctions associées		Détails
Prise en main facile	Pour tous	Trouver l'application facilement	Application native : icône et nom intuitifs
	Pour les participants	Utilisation intuitive	Travailler sur l'ergonomie et la structure
		Affichage précis et pertinent	Obtenir les bonnes informations au bon endroit
		Faciliter l'accès aux informations	Rapidité d'exécution
	Pour les VP Tournoi	Supprimer les intermédiaires	Entrée des scores dynamique
		Obtenir un espace dédié	Créer des comptes VP Tournoi et un espace ergonomique pour l'entrée des scores
Un contenu qui couvre l'ensemble de l'événement	Diffuser les photos		Créer une galerie photos (base de données)
	Diffuser les news		Créer un newsfeed
	Diffuser la programmation des matchs	Le lieu & les horaires	Afficher les informations en amont du tournoi
		Les résultats	Pouvoir mettre à jour en temps réel les résultats
	Proposer un classement des écoles		Créer un onglet classement
	Donner accès à des informations générales		Créer un espace dédié aux informations
Une application dynamique et pérenne	Dynamisme	Pages statiques	Les créer si la modification d'information nécessite une mise à jour de l'application
		Pages dynamiques	Pour toutes les infos qui doivent se mettre à jour en temps réel
	Pérennité		Mettre en place une base de données complète

TABLE 1.1 – Synthèse du cahier des charges

## Partie 2

# État de l'art et bilan des connaissances : choix techniques

### 2.1 Choix de la méthode : comment créer une application ?

La méthode est simple et invariante.

- Le cahier des charges fournit les fonctionnalités de l'application. Une priorisation de ces fonctionnalités est effectuée et un modèle précisant les outils qui seront utilisés, les méthodes suivies ainsi que le planning des tâches est mis au point.
- On « écrit » l'application sur son ordinateur, c'est-à-dire qu'on la code avec un certain langage suivant la plateforme visée ou le type d'application que c'est. L'application est composée de plusieurs fichiers.
- On envoie l'application aux distributeurs (Apple, Google) qui l'analyseront avec plus ou moins de minutie.
- Une fois validée, elle est disponible dans les boutiques virtuelles (AppStore ou PlayStore).

Trois scénarios de développement sont alors envisageables pour la conception d'applications mobiles (partie « écriture » de l'application) :

- 1) La programmation native propose de faire usage des langages spécifiques à chaque système d'exploitation, tels que Swift pour iOS ou Java pour Android. C'est le plus bas niveau de codage, donc celui qui offre une totale liberté pour le développeur.
- 2) La multiplication des forfaits data a également donné naissance aux services web mobiles codés en HTML et pour lesquels l'utilisateur devra pointer son navigateur mobile vers une URL spécifique. Autrement dit, il s'agit de concevoir une application web pour qu'elle soit compatible avec les navigateurs des smartphones. C'est le plus haut niveau de codage, ce qui limite rapidement la conception d'applications mobiles.
- 3) Il existe enfin les applications hybrides développées majoritairement en HTML et compilées avec des frameworks pour obtenir des applications natives avec plus ou moins certaines différences par rapport à la programmation native. C'est un niveau de codage intermédiaire, qui combine les deux mondes.

La conception d'une application native est accordée si les langages de programmation sont connus. Même s'ils sont Open Source, Swift pour iOS et Java pour Android ne font pas partie des langages les plus utilisés par les étudiants.

Apprendre le fonctionnement de ces langages, leurs bibliothèques, leur paradigme demande beaucoup de temps et d'expérience pour les maîtriser totalement.

Notons aussi que le travail est double : il existe deux plateformes (iOS et Android) donc deux langages, donc deux développements en parallèle.

Nous avons donc d'un commun accord décidé d'opter pour la troisième solution présentée : la production d'une application hybride, basée sur des solutions web (HTML, CSS, PHP : des langages déjà plus ou moins connus) mais qui utilisera des Frameworks dédiés au développement mobile.

## 2.2 Choix du logiciel

Vu que le choix de développement de notre application porte sur une application hybride, il faut choisir les outils les plus performants et une architecture adaptée à sa réalisation. C'est pourquoi les outils suivants ont été choisis.

### **Ionic Framework**

Un Framework fournit des fonctionnalités standards et impose un cadre. Il faut donc que le code développé soit de qualité et structuré de façon à être évolutif.

Ionic Framework est un mélange d'outils et de technologies pour développer des applications mobiles hybrides rapidement et facilement. Il s'appuie sur AngularJS pour la partie application web du framework et sur Cordova pour la partie construction des applications natives.

Ce framework Open Source permet de développer une application déployable sur plusieurs environnements tel qu'un site web ou une application mobile pour des systèmes tel que iOS ou Android.

En plus de cela, il constitue l'un des Frameworks les plus utilisés actuellement grâce à une communauté active et une documentation assez riche.

Ce framework est basé sur une architecture MVC (Model-View-Controller).



FIGURE 2.1 – Ionic

## Phalcon

Phalcon est un Framework PHP basé sur le patron de conception MVC et qui met en avant des performances élevées.

Contrairement à la majorité des frameworks PHP, Phalcon est implémenté sous la forme d'une extension écrite en langage C afin d'optimiser les performances. Cette démarche permet d'améliorer la vitesse d'exécution tout en diminuant la consommation de ressources et a pour objectif de permettre la gestion d'un plus grand nombre de requêtes par seconde par rapport à des frameworks comparables écrits en PHP. [8]

## MySQL

C'est un Système de Gestion de Bases de Données Relationnelles (SGBDR), c'est-à-dire un logiciel qui permet de gérer des bases de données, et donc de gérer de grosses quantités d'informations. Il utilise pour cela le langage SQL. Il s'agit d'un des SGBDR les plus connus et les plus utilisés (Wikipédia et Adobe utilisent par exemple MySQL). [9]

## 2.3 Choix de l'architecture : architecture MVC

L'architecture MVC (Model-View-Controller) (Figure 2.2) est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective.

Un programme construit selon ce modèle est donc séparé en trois parties.[10]

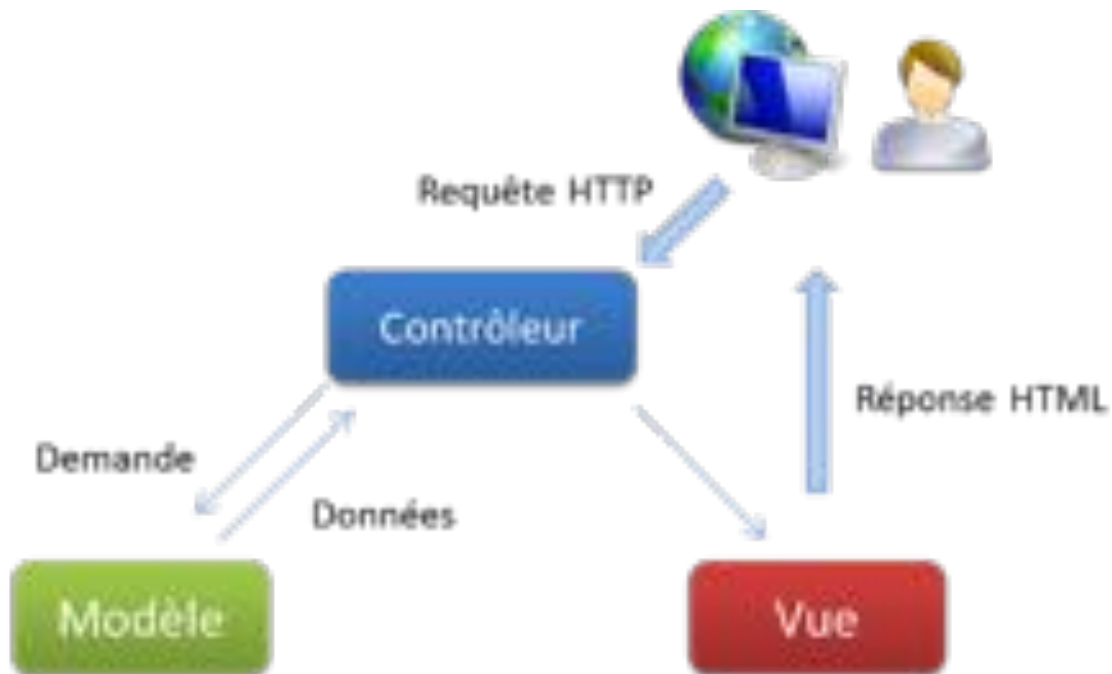


FIGURE 2.2 – Modèle MVC

### Model (Modèle)

Le modèle représente le cœur (algorithmique) de l'application : traitements des données, interactions avec la base de données... Il décrit les données manipulées par l'application.

### View (Vue)

Ce avec quoi l'utilisateur interagit se nomme précisément la vue. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toute action de l'utilisateur (hover, clic de souris, sélection d'un bouton radio, cochage d'une case, entrée de texte, de mouvements, de voix, etc.). Ces différents événements sont envoyés au contrôleur. La vue n'effectue pas de traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle et d'interagir avec l'utilisateur.

### Controller (Contrôleur)

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de la vue et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle afin que les données affichées se mettent à jour.

## 2.4 Architecture de l'application

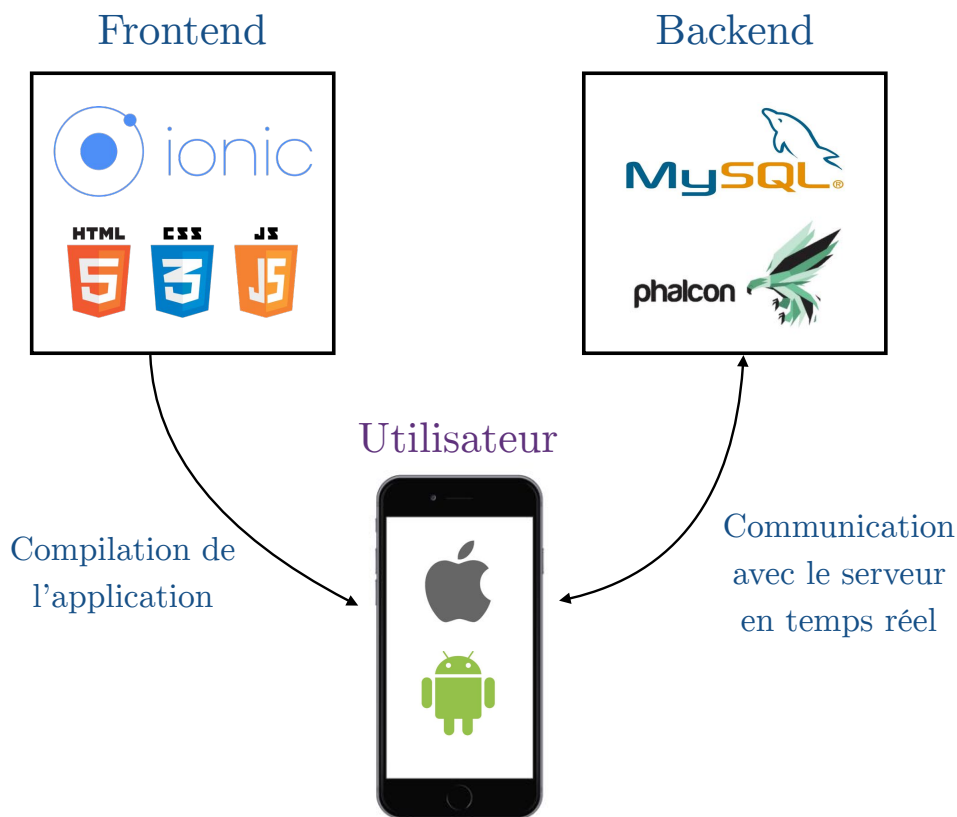


FIGURE 2.3 – Architecture globale



## Partie 3

# Ouverture du projet et reflexion sur l'application

### 3.1 Mise en place du frontend

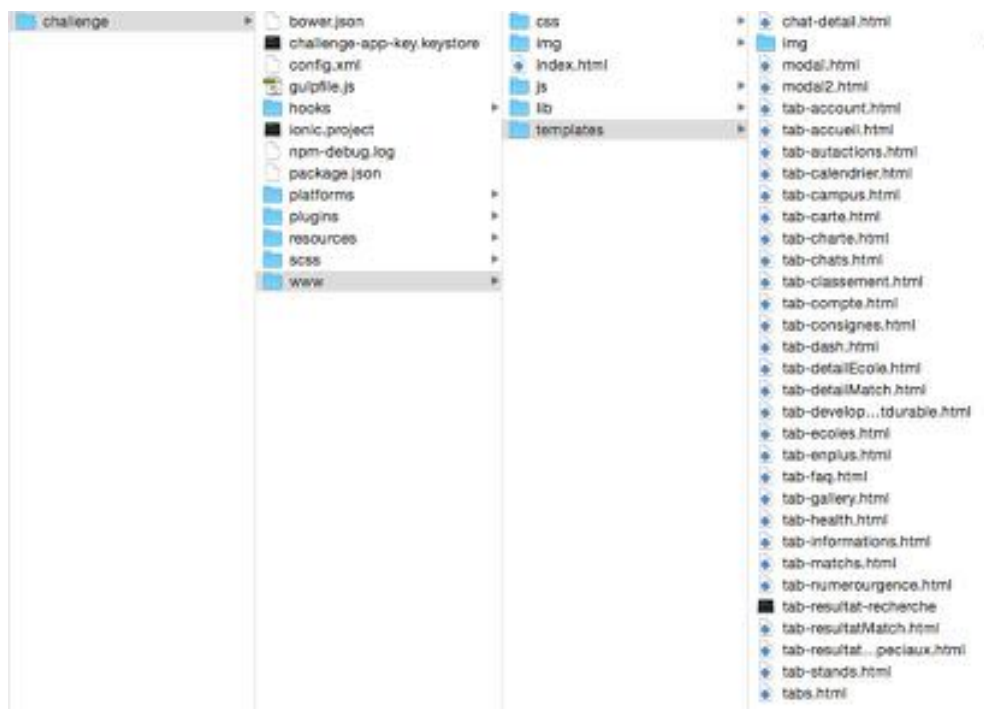


FIGURE 3.1 – Fichiers de l'application

Pour pouvoir développer le frontend, il nous faut installer ionic. Ionic produira ensuite un projet de base avec un certain nombre de fichiers. Voici la méthode à suivre, en lignes de commande :

```
$ sudo apt-get install nodejs      $ ionic start challenge tabs
$ sudo apt-get install npm         $ cd challenge
$ npm install -g cordova ionic     $ ionic serve
```

A partir de là, le projet (challenge) est créé. Il est alors lancé en local.

La Figure 3.1 montre l'ensemble du projet. Ionic crée automatiquement les fichiers et dossiers de la deuxième colonne. Il nous suffit alors de les remplir avec des fichiers que nous créeront afin de fabriquer notre application.

## 3.2 Mise en place du backend

### Acquisition et mise en place du serveur

Le serveur qui héberge le backend est une machine virtuelle (VM) avec un système d'exploitation *debian8*. Cette VM est hébergée sous le nom *challenge2016* dans une machine physique (Artemis) chez ÉCLAIR. L'adresse IP publique de cette VM est **challenge2016.eclair.ec-lyon.fr** (ou 156.18.24.133).

La procédure d'installation d'une VM sur les serveurs d'ÉCLAIR étant très long à décrire ici, tous les détails pourront être trouvés dans le wiki d'ÉCLAIR.

### Installation de MySQL, PHP et Apache2

Par défaut, lors de son installation, Debian propose d'installer certains logiciels ou services de bases. Ainsi, le serveur MySQL et Apache peuvent être installés à cette étape si l'utilisateur a accepté la proposition de debian. Sinon, à la fin de l'installation de debian, voici la commande qui permet de les installer [11] :

```
$ sudo apt-get install mysql-server-5.5 apache2 php5-mysql
```

### API Rest

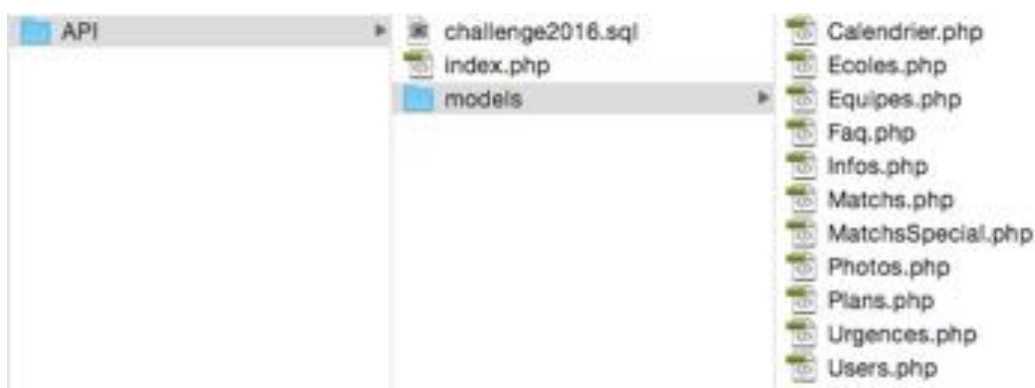


FIGURE 3.2 – Fichiers de l'API

Conçu avec le Framework Phalcon, cet API (Figure 3.2) assure le fonctionnement dynamique de l'application. Le fichier principal (le plus important donc) de cet API est *index.php* qui contient l'ensemble du programme permettant de réaliser toutes les requêtes provenant de l'application mobile.

### 3.3 Vue globale de l'application

Après avoir étudié le cahier des charges ainsi que les outils techniques à notre disposition nous présentons en Figure 3.3 le modèle de notre application :

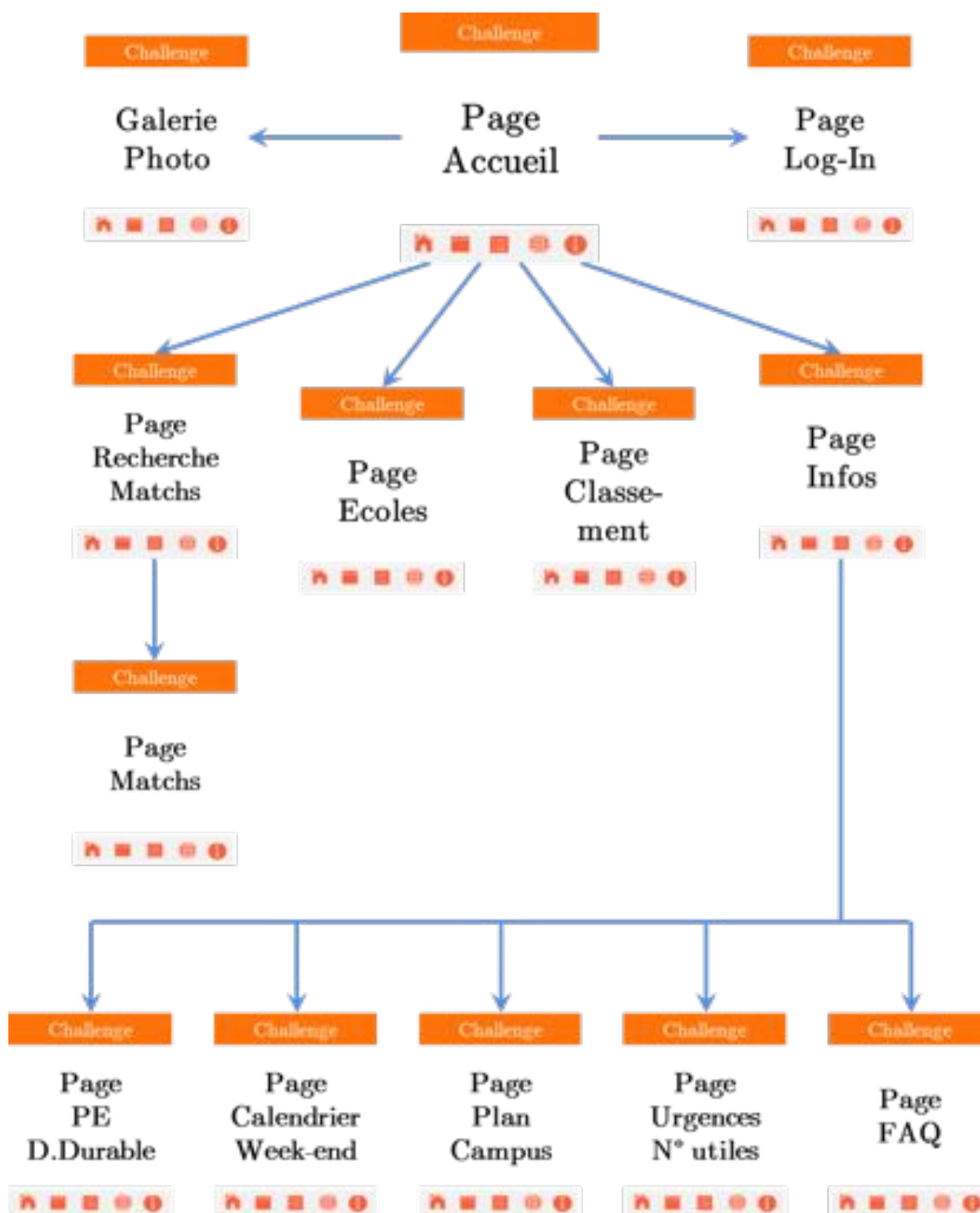


FIGURE 3.3 – Modèle de l'application

### 3.4 Gestion de la base de données de l'application

Pour résumer les différentes étapes du projet (voir Figure 2.3), sont créés pour le moment :

- le frontend ;
- le backend (serveur + base de données + API).

A priori, l'application est prête à être codée, puis envoyée : en effet si le frontend et le backend sont terminés, ionic se charge de compiler l'application pour Apple et Google et après validation par les deux distributeurs, elle pourra être téléchargée sur l'AppStore et le PlayStore. Sauf qu'à ce moment là, l'application est vide. Elle est certes fonctionnelle, mais vide, c'est-à-dire sans aucune information à afficher. En effet, la base de donnée SQL est vide, elle n'a pas été remplie.

Mais pour entrer toutes les informations dans la base de données (écoles, participants, programmation des matchs...) il est absolument inconcevable d'écrire des requêtes SQL et d'entrer les information « à la main ». Rappelons que le Challenge fait intervenir près de 3000 participants, chacun concourant dans une épreuve sportive. Des milliers de matchs vont être joués ce week-end avec à chaque fois autant d'informations à entrer (horaires, équipes, lieu...).

Pour respecter l'aspect important du cahier des charges (Tableau 1.1) qui est la pérennité de l'application, il fallait mettre en place un outil de gestion de notre base de données efficace. Le but de cet outils et de faciliter le remplissage de la base de données, tout en ayant une représentation claire de sa structure. Nous avons donc créé une web application, spécialement conçue pour la gérer.

La page de présentation de cette web application (un site internet accessible à l'adresse <http://challenge-2016.eclair.ec-lyon.fr/Formulaire/>) est représentée ci-dessous (Figure 3.4) :

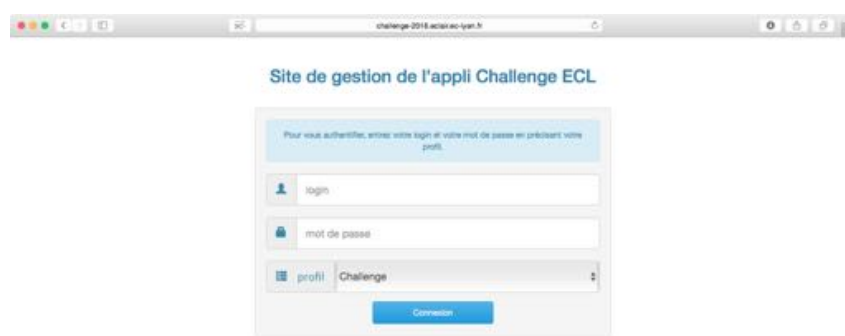


FIGURE 3.4 – Page d'accueil du site internet permettant la gestion de la base de données de l'application

Il suffit alors de se connecter avec des identifiants spécialement conçus et nous pouvons alors gérer notre base de données.

L'architecture globale finale du projet est alors décrite Figure 3.5 :

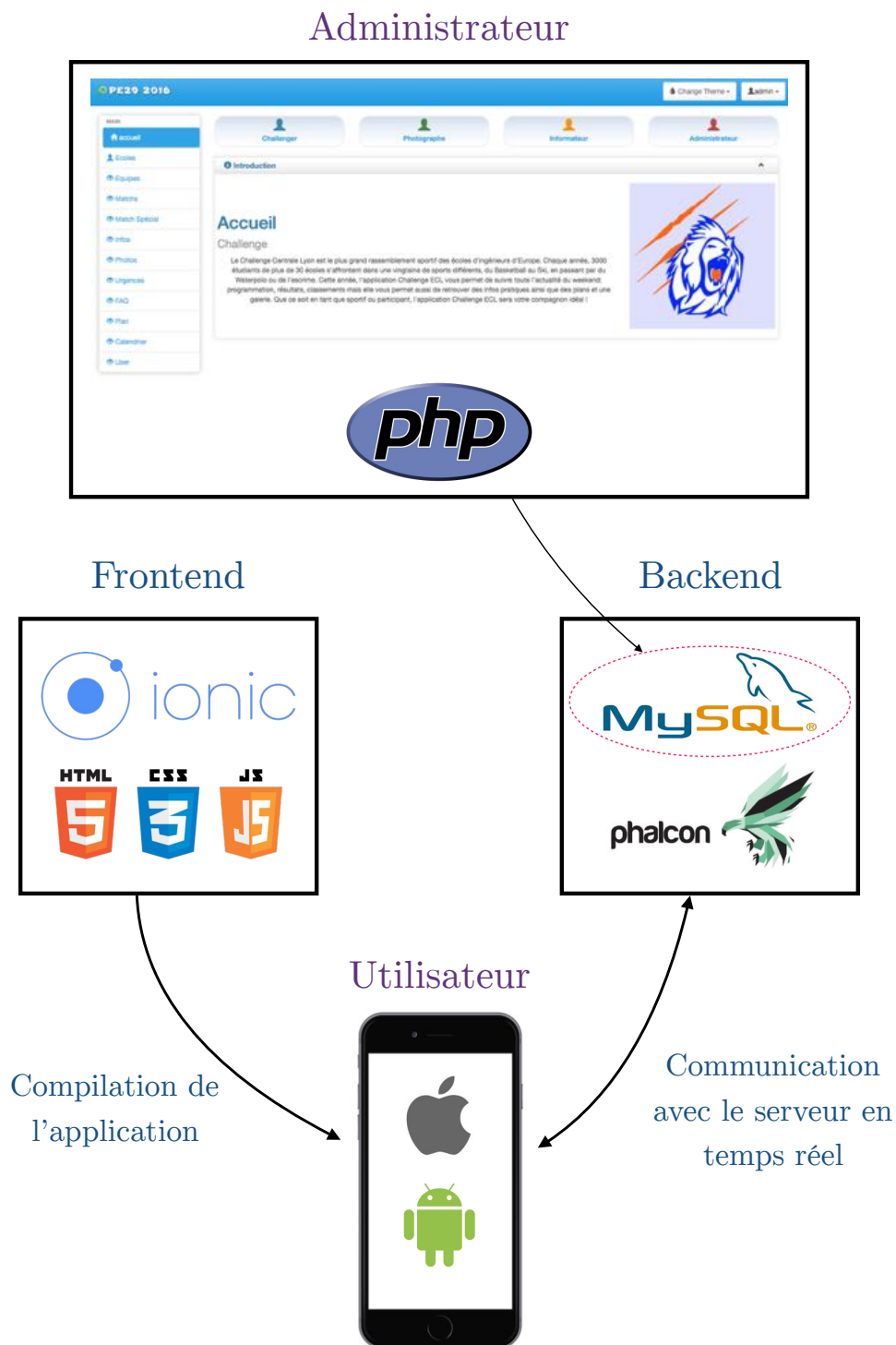


FIGURE 3.5 – Architecture globale finale

### 3.4.1 Architecture de la base de données

Avant d’être implémentée dans un SGBD, voire même être codée avec MySQL, une base de données doit être impérativement modélisée et étudiée dans les moindres détails. En effet, la base de données constitue la brique de base de tout système informatique dynamique.

Dans le cas de notre application, cette modélisation a été faite avant le début du codage. Pour cela, on établit le schéma relationnel qui relie les différentes tables de notre base de données ainsi que leur contenu.

Pour l’implémentation, on a utilisé MySQL comme SGBD et la base a été nommée *challenge2016*. Elle comprend 11 tables qui sont nommées ainsi (Tableau 3.1) :

challenge2016
Calendrier
Ecoles
Equipes
Faq
Infos
Matches
MatchesSpecial
Photos
Plans
Urgences
Users

TABLE 3.1 – Base de données *challenge2016*

La structure interne de chaque table est accessible depuis le fichier *challenge2016.sql* qui accompagne le dossier contenant les codes sources. Pour des raisons de sécurité, les mots de passe et autres détails d’accès ne sont pas divulgués dans ce rapport.

Cela signifie aussi que toutes les informations qui sont présentes dans la base de données sont donc considérées comme « dynamiques ». Elles peuvent être mises à jour, modifiées, supprimées indépendamment par l’administrateur.

Évidemment, les tables Matches (pour les sports collectifs) et MatchesSpecial (pour les sports individuels) devront être modifiées tout au long du week-end. Même traitement pour la table Photos qui est supposée dynamique par construction.

Page suivante est représentée le schéma relationnel de la base de données. Il met en évidence les liens qui existent entre les tables et les informations utiles.

### 3.4.2 Schéma relationnel de la base de données

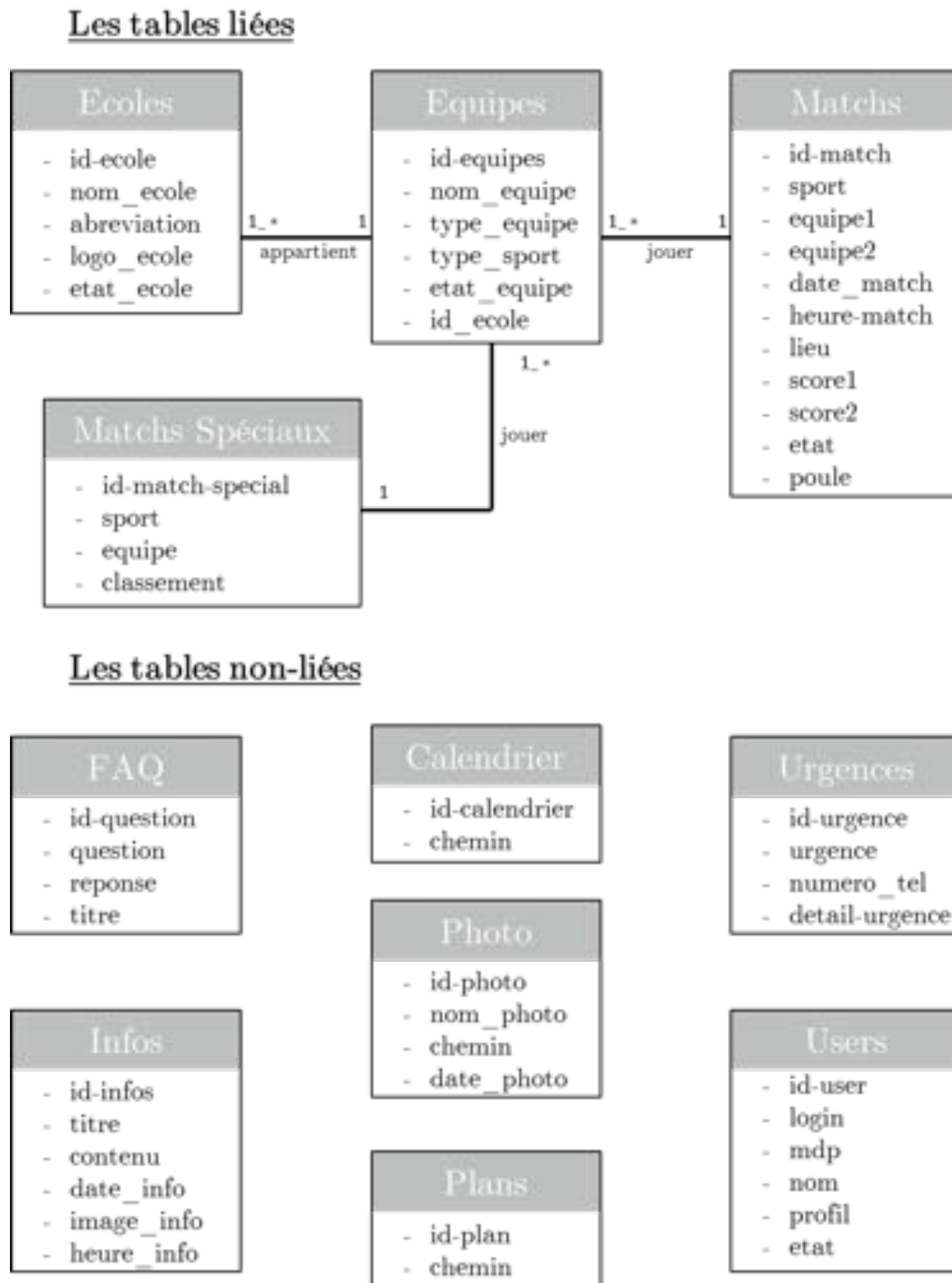


TABLE 3.2 – Schéma relationnel de la base de données

## 3.5 Travail sur l'expérience utilisateur (User Experience)

Une application se doit d'être intuitive, simple (attention pas simpliste), agréable à utiliser et à regarder. Un travail de structure doit être mené en amont pour amener l'expérience utilisateur au plus haut niveau.

L'objectif de l'application est de donner facilement et rapidement accès aux informations du week-end. Les matchs et leurs résultats qui sont le centre d'intérêt de l'événement devaient être accessibles de manière intuitive et rapide. Dès lors, pour limiter le nombre de clic de l'utilisateur plusieurs options ont été choisies.

### *Le choix des onglets*

Tous les onglets de l'application devaient avoir un poids similaire au sein de l'application. Ce poids ne se calcule pas en quantité d'information mais en l'intérêt qu'un individu peut y porter. En effet, la page classement n'est composé que d'une unique information mais reste pourtant un onglet à part entière car tout l'intérêt d'une compétition sportive réside dans le classement final. Enfin, la place discrète du lien vers la page de log-in est volontaire car elle ne concerne que les organisateurs (et VP Tournois) et n'a aucune importance pour un simple participant.

### *L'utilisation d'un menu déroulant*

Afin de favoriser la rapidité d'accès aux informations, il était important de minimiser le nombre d'actions (clics sur l'écran) de l'utilisateur. Dès lors, pour les pages Matches et Accueil qui comportent une importante quantité d'information, le menu déroulant était idéal. En effet, il évitait d'avoir à créer de nouvelles pages pour afficher les résultats des matchs et permettait également d'accéder rapidement à toutes les parties de la page donc à toutes les informations.

### *Le design*

Un design simple et épuré permet de concentrer l'attention des utilisateurs sur le contenu, sans que ce soit désagréable. L'information est recentrée et donc appréciée à sa juste valeur. Le choix de la charte des couleurs a été long et controversé mais le choix a du être fait et peu à priori évoluer pour une prochaine mise à jour.

## 3.6 Total des lignes de code

En tout ce sont :

- 3181 lignes de code pour le frontend ;
- 2337 lignes de code pour le backend ;
- 220 lignes de code pour le design (CSS).

Soit un total de 5378 lignes de code pour les 3 parties principales. Les légers ajustements dans d'autres fichiers ne sont pas comptés ici.



## Partie 4

# Une application mobile inédite : Challenge ECL

### 4.1 Télécharger l'application

L'application est téléchargeable sur les deux plus grandes plateformes de distribution du monde (Figure 4.1 : l'AppStore d'Apple (système iOS) et le PlayStore de Google (système Android). Les iPhones équipés d'iOS et les smartphones équipés d'Android représentent 90,4% des smartphones en circulation dans l'Hexagone. [12]

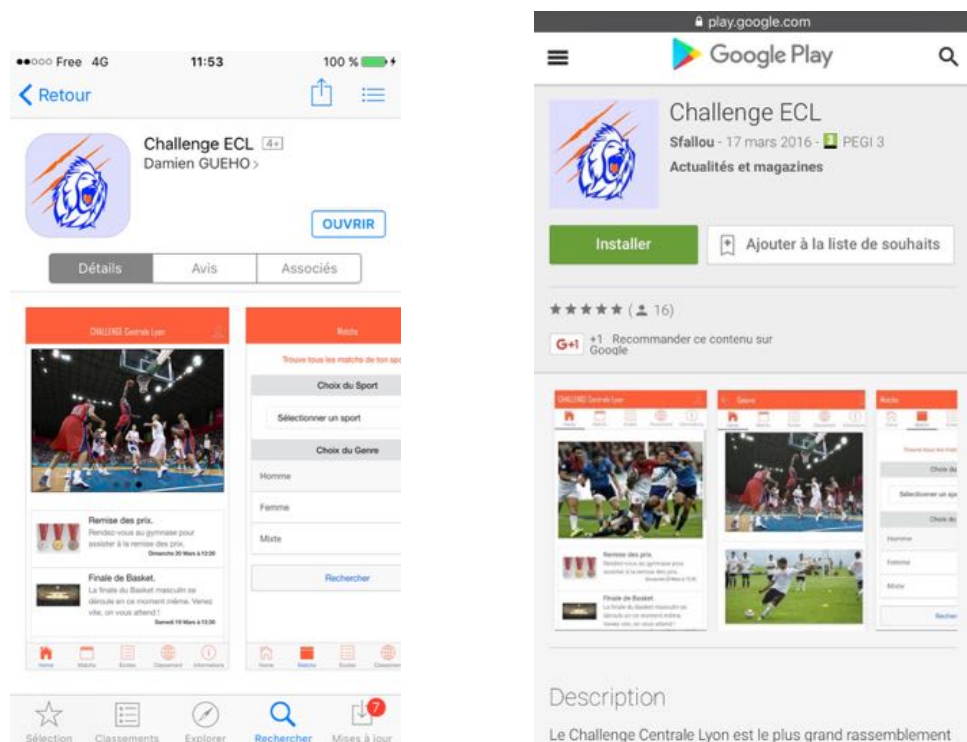


FIGURE 4.1 – Challenge ECL sur l'AppStore et le PlayStore

## 4.2 Présentation de l'application

Classiquement, notre application est munie de « tabs » (en bas sur iPhone et en haut sur Android) qui permettent d'accéder aux différentes fonctionnalités proposées (Figure 4.2) :

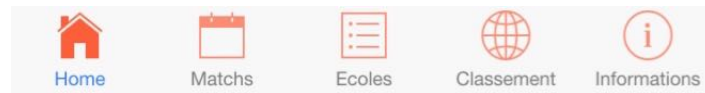


FIGURE 4.2 – Les *tabs* de l'application

Chaque *tab* permet d'accéder à une page.

### 4.2.1 Home

L'application se doit d'être vivante, attirante, dynamique afin de pousser l'utilisateur à s'en servir souvent. Dès lors, il est impératif d'avoir une page d'accueil avec un fort contenu (Figure 4.3).



FIGURE 4.3 – Page d'accueil de l'application

La page d'accueil se décompose en plusieurs parties :

- comme l'ensemble des pages, elle est constituée d'une barre fixe présentant les tabs (liens vers les autres pages) ;
- en haut à droite, un bouton pour accéder à l'interface de connexion des VP Tournois et les organisateurs ;
- un emplacement en haut dédié aux trois photos du moment (les dernières ajoutées dans la galerie) qui défilent en boucle ;
- un Newsfeed présentant une série de News (Annexe C) indiquant les différents temps forts ou moments importants (comme l'ouverture du RU par exemple) du week-end.

## 4.2.2 Galerie

On accède à la Galerie (Figure 4.4) en tapant sur la photo sur la page d'accueil.

La Galerie photos possède une présentation assez simple sous la forme d'une liste de photos. Cette page présente un intérêt tout particulier pour les personnes qui souhaitent pouvoir suivre le déroulement de l'événement même s'ils ne sont pas sur les lieux des matchs.

On peut aussi zoomer sur les photo en tapant dessus.

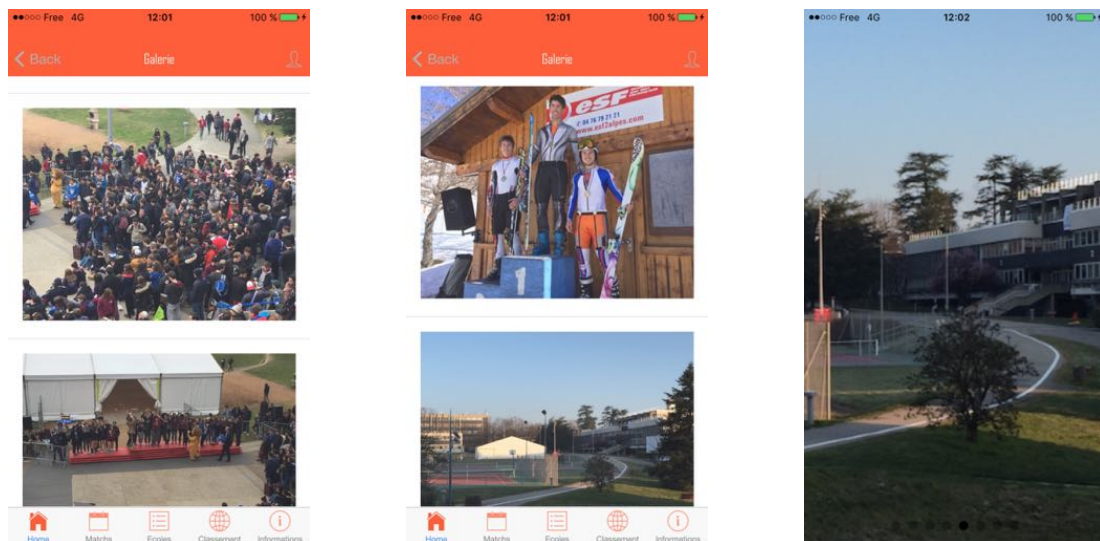


FIGURE 4.4 – Galerie photos ; la dernière image représente une image zoomée

## 4.2.3 Espace connexion

Le Challenge représente une colossale organisation logistique des tournois, ce qui demande beaucoup de communication entre tous les organisateurs. La possibilité qu'ont les VP Tournois de transmettre les résultats directement depuis l'application est alors une aide non-négligeable. Pour ce faire, les VP Tournois se connectent à partir de l'interface de connexion (Figure 4.5) accessible depuis la page d'accueil.

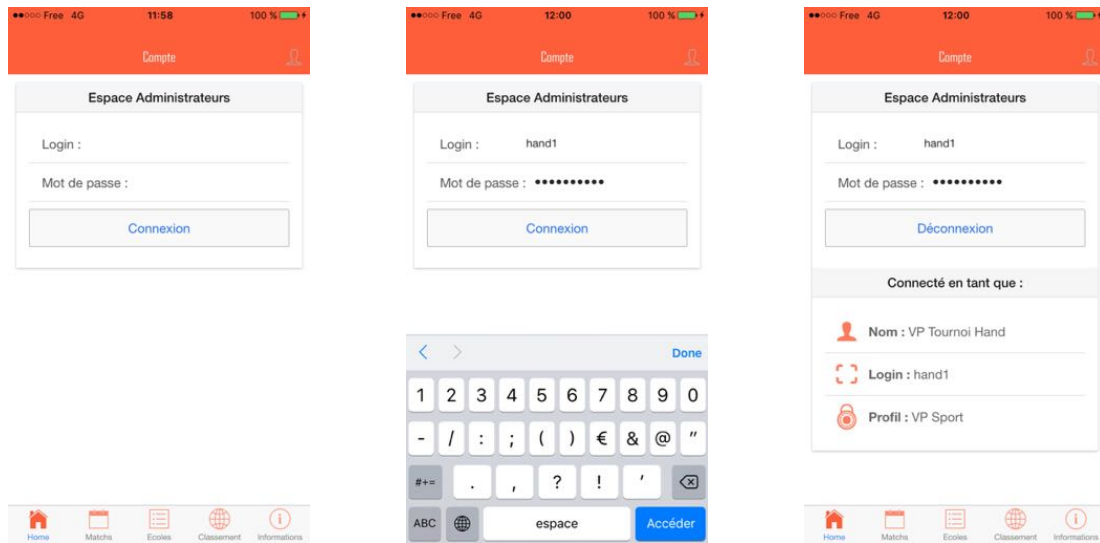


FIGURE 4.5 – Espace connexion

Une fois connecté, il est possible de modifier les scores de la page match directement depuis son smartphone comme on peut le voir dans la partie suivante.

#### 4.2.4 Matches

Rappelons que le Challenge est avant tout un événement sportif. Il est donc important d'avoir accès facilement à l'ensemble de la programmation de tous les tournois du week-end.

Dans un premier temps, l'utilisateur doit préciser la recherche qu'il souhaite effectuer (Figure 4.6) en sélectionnant un sport et un genre.

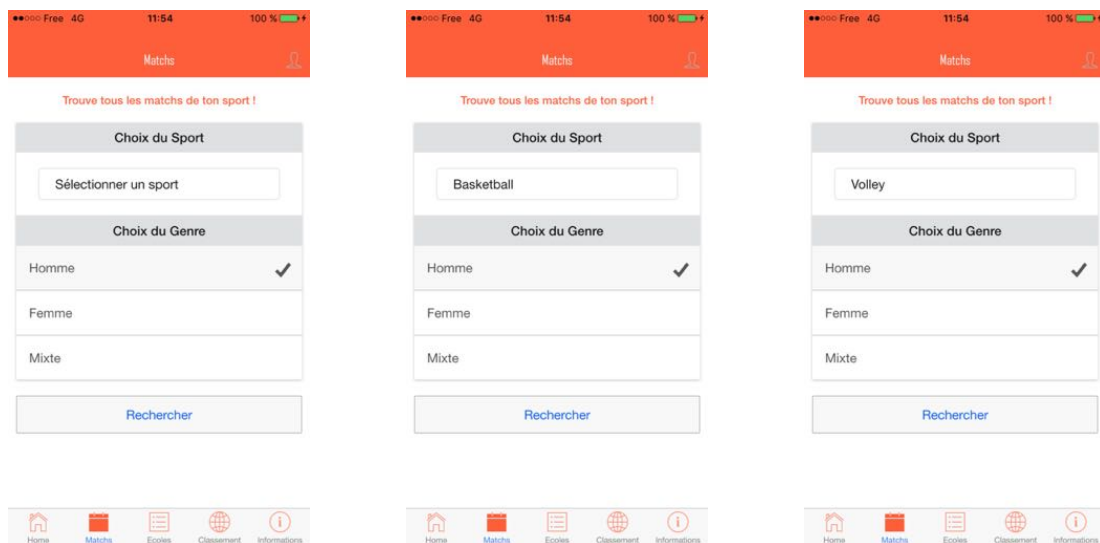


FIGURE 4.6 – Page Matches : sélection d'un sport et d'un genre

Une fois sa recherche effectuée, l'utilisateur tape sur « Rechercher » et arrive sur une nouvelle page. Plusieurs exemples sont donnés ci-dessous (Figures 4.7, 4.9 et 4.8) :

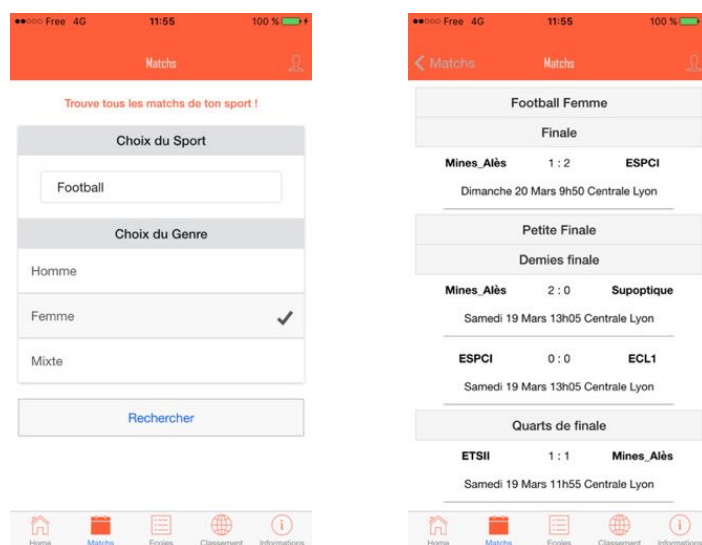


FIGURE 4.7 – Page Matches : recherche du tournoi de football féminin

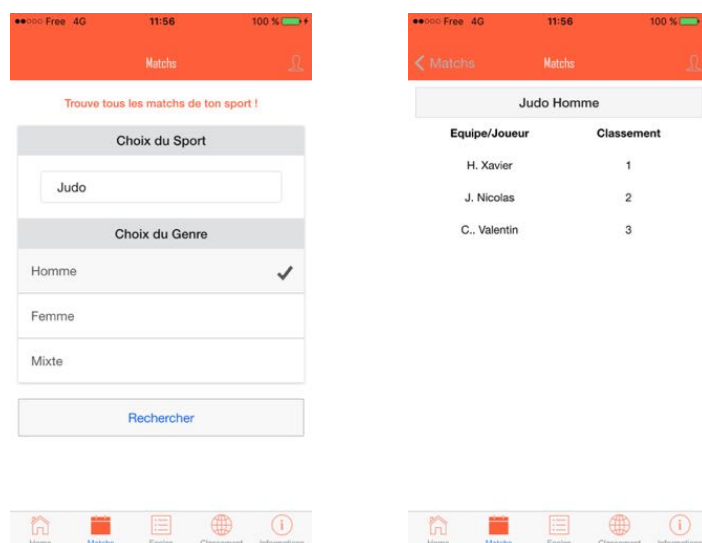


FIGURE 4.8 – Page Matches : recherche du tournoi de judo masculin

Lorsque le tournoi se joue en individuel (comme par exemple le judo, le tennis, la natation...), l'application affiche le classement des participants, puisqu'il n'existe pas de programmation « claire ». Du moins, il n'est pas très pertinent d'afficher les centaines de matchs de tennis qui seront joués par exemple. Seul le classement final compte. La liste est donnée dans le Tableau 4.1.

Sports avec affichage d'une programmation complète	Sports avec affichage seulement du classement final
Basketball	Athlétisme
Football	Badminton
Handball	Escalade
Rugby	Escrime
Volleyball	Golf
	Judo
	Natation
	Raid
	Ski
	Tennis
	Tennis de table
	Ultimate
	Warerpolo

TABLE 4.1 – Méthodologie d’affichage des sport

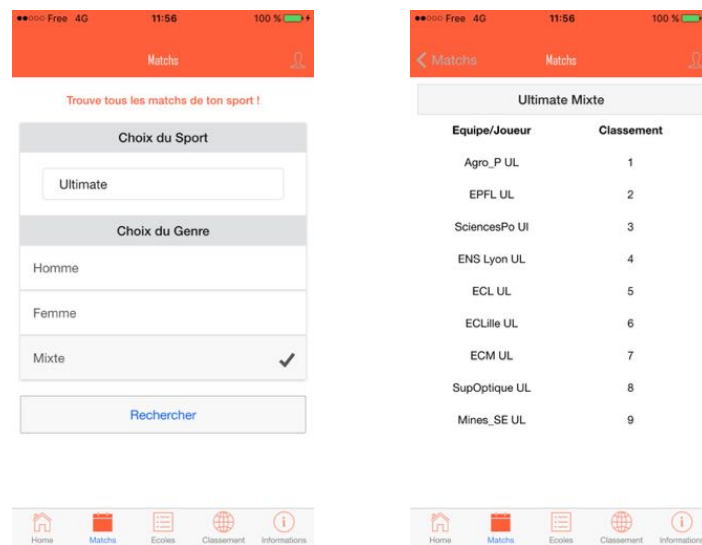


FIGURE 4.9 – Page Matches : recherche du tournoi d’ultimate (tournoi mixte)

Lors du Challenge, dans chaque discipline, une multitude de matchs sont joués. Il faut donc pouvoir les présenter de manière intelligente pour que l'utilisateur puisse facilement s'y retrouver. Pour cela, la présentation s'est faite de la façon suivante : en haut sont placés les matchs de phases finales (par ordre d'importance), puis plus bas dans la page arrivent les phases de poules, comme le montre par exemple la programmation du football féminin (Figure 4.10).

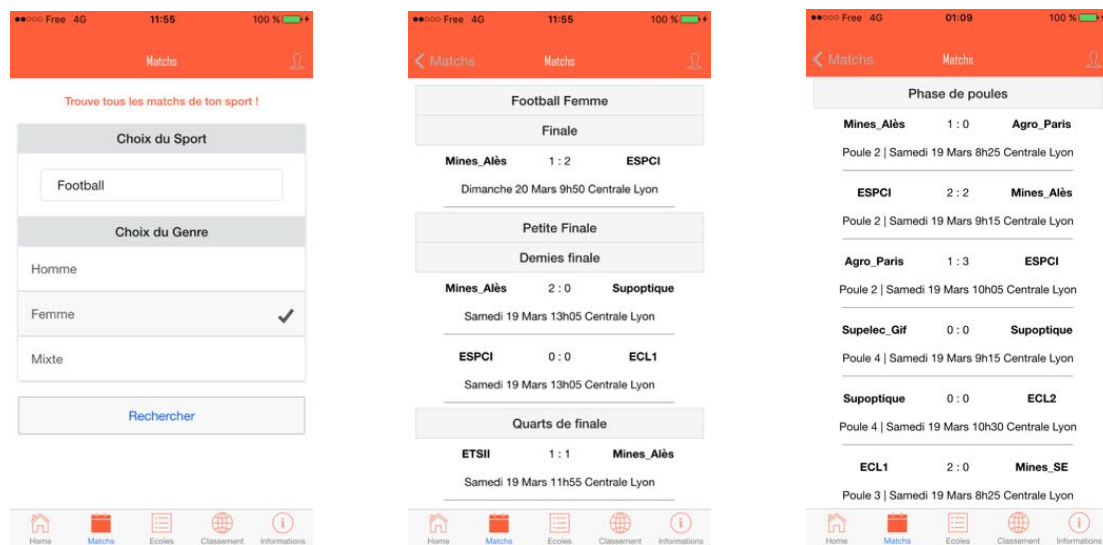


FIGURE 4.10 – Football féminin : phases de poules en bas et phases finales en haut

Si l'utilisateur est un VP Tournoi et s'est connecté avec ses identifiants, ce dernier verra une légère différence sur la page Matches. En effet, comme le montre la Figure 4.11, il lui est possible de modifier le score d'un match ainsi que les équipes composant ce match. Ainsi, il peut gérer lui-même le tournoi dont il est responsable.

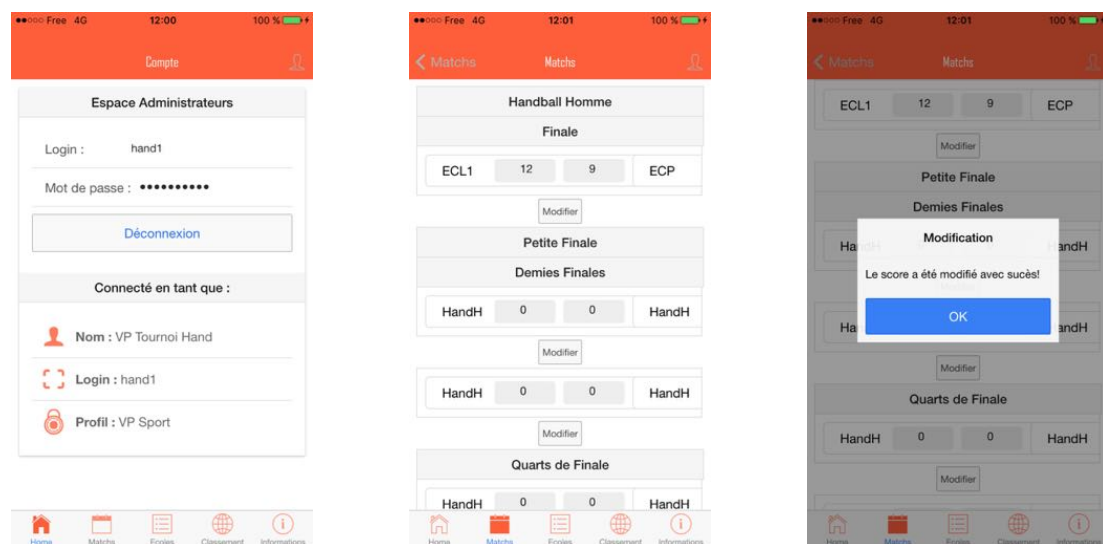


FIGURE 4.11 – Interface de la page Matches pour un VP Tournoi connecté (en Handball ici)

### 4.2.5 Ecoles

Le Challenge est une compétition sportive où 34 écoles d'ingénieurs participent. Chaque participant est aussi présent pour soutenir sa propre école et doit pouvoir suivre l'évolution de son école dans la compétition. C'est là que cette page intervient.

Après avoir sélectionné l'école souhaitée, il est possible d'avoir accès aux points que possède l'école (et son nombre de points dans le classement) ainsi qu'à l'ensemble des matchs joués et qui restent à jouer pour l'ensemble des participants de l'école (Figure G.8). L'utilisation de cette page est extrêmement simple et donne un accès rapide aux informations cherchées.

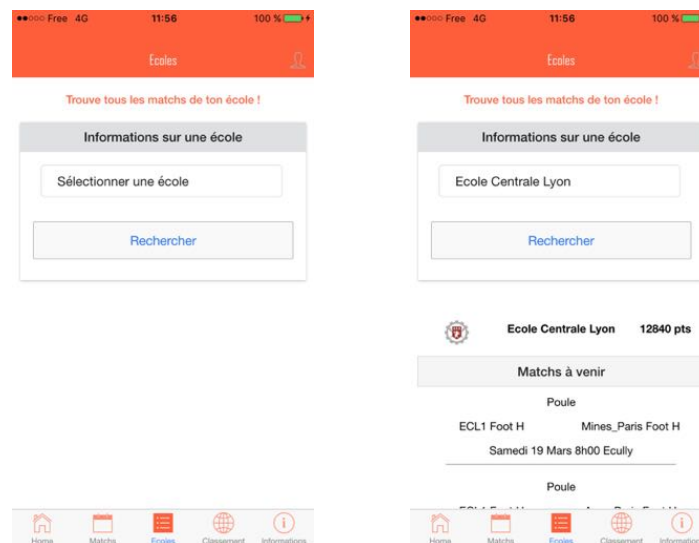


FIGURE 4.12 – Page Ecoles

### 4.2.6 Classement

L'intérêt d'une compétition sportive est bien d'obtenir un classement entre tous. C'est pour cette raison que cette page existe (Figure G.5). Grâce à un algorithme écrit par le Challenge lui-même, un classement est donné en fonction des résultats de chaque école dans chaque sport.

#### *Méthodologie de l'algorithme*

200 points sont attribués pour chaque sport :

- 100 points pour l'école arrivant à la première place ;
- 60 points pour l'école arrivant à la deuxième place ;
- 40 points pour l'école arrivant à la troisième place.

Pour chaque sport, le nombre de point est ensuite multiplié par le ratio :

$$\rho = \frac{\text{nombre de participants dans le sport considéré}}{\text{nombre de participants au Challenge}}$$



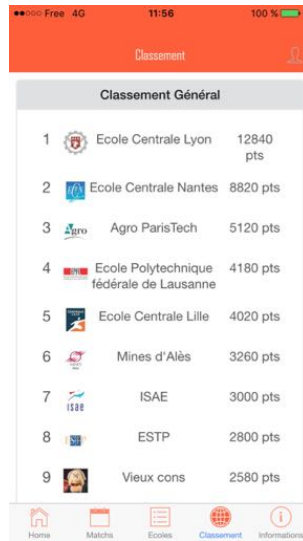


FIGURE 4.13 – Page Classement

## 4.2.7 Informations

La page Informations contient toutes les informations nécessaires pour passer un bon week-end. De là, on peut accéder aux sections (Figure 4.14) :

- développement durable ;
- calendrier du week-end ;
- plan du campus ;
- urgences : numéros utiles ;
- FAQ.



FIGURE 4.14 – Page Infos

## Développement durable

Notre groupe de PE était chargé d'inclure dans l'application une partie du travail du PE 90 concernant le développement durable au Challenge. Voici Figure 4.15 des exemples d'informations que l'on peut trouver dans la section développement durable.

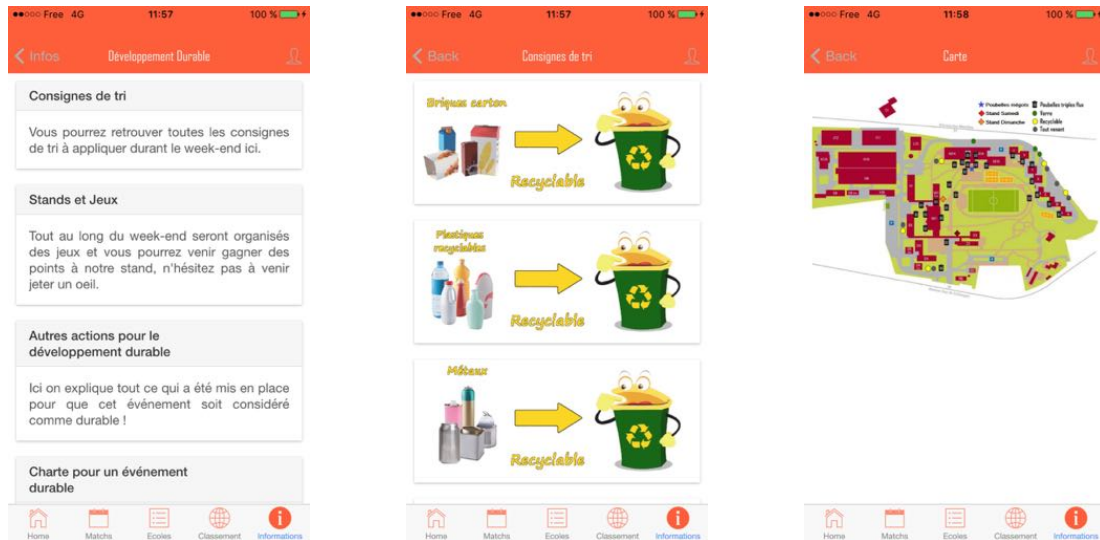


FIGURE 4.15 – Section développement durable de la page Infos

## Calendrier du week-end

On trouve ici le calendrier du week-end mis à jour suivant la date à laquelle on le consulte (Figure 4.16).



FIGURE 4.16 – Ici, programme du Dimanche

## Plan du campus

Pour se repérer sur les campus et les différents sites sportifs, les participants avaient accès à plusieurs cartes fournies directement par le Challenge (Figure 4.17). Il y a la possibilité de zoomer sur les cartes.

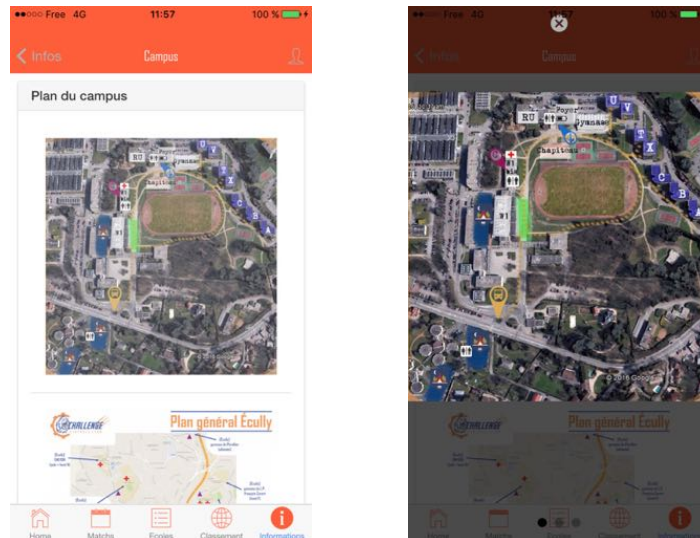


FIGURE 4.17 – Plan du campus (affichage normal puis zoomé)

## Urgences : numéros utiles

En cas d'urgence, les participants avaient la possibilité de joindre la Croix-Rouge ou un des membres du Challenge (Figure 4.18).

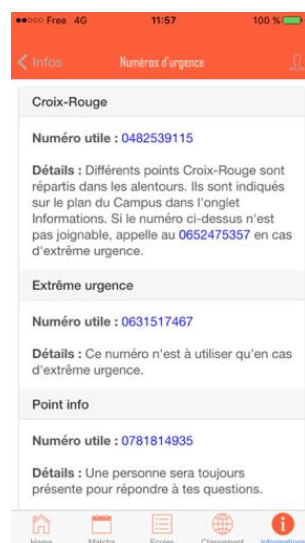


FIGURE 4.18 – La section urgences avec des numéros utiles

## FAQ

La FAQ (Foire Aux Questions) est un endroit où l'on trouve beaucoup de réponses aux questions usuelles. Elle contient l'ensemble des informations utiles pour les participants en cas d'interrogation. Elle est organisée en pôles (Sport, Communication, Soirée, Sécurité...). En voici quelques aperçus Figure 4.19 :

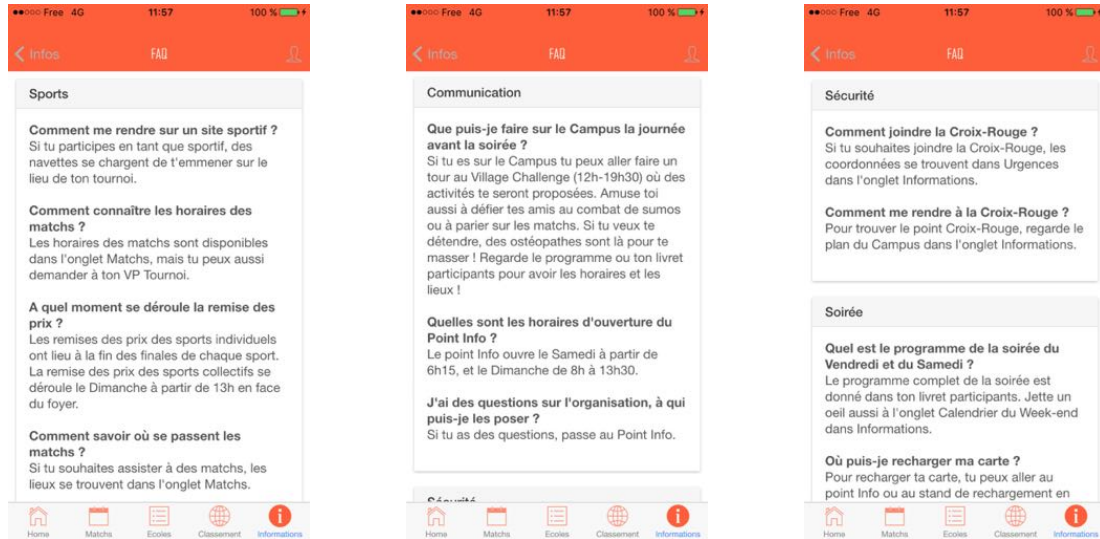


FIGURE 4.19 – FAQ

## Partie 5

# Dispositifs de communication

### 5.1 La communication externe à l'École

#### 5.1.1 Facebook

Une fois l'application en libre téléchargement sur l'AppStore et le PlayStore, il est temps de la promouvoir pour inciter les participants à la télécharger. Avant tout, Facebook, qui est aujourd'hui le réseau social incontournable pour tous les étudiants, a été notre première cible. Une page pour l'application (Figure 5.1) a alors été créée pour pouvoir partager le lien de téléchargement.



FIGURE 5.1 – Page Facebook

#### 5.1.2 Vidéo

Nous avons aussi décidé de se servir de la page facebook du Challenge. En effet, le Challenge proposait en amont de l'événement la possibilité de poster chaque jour (à partir de J-20) une courte vidéo afin de montrer aux participants la préparation de l'événement. Ainsi, une vidéo parodique des produits Apple apparaissait comme idéal pour promouvoir l'application auprès de tous. 7 jours avant le Challenge, nous proposons notre vidéo sur internet. Figure 5.2, on voit quelques aperçus.

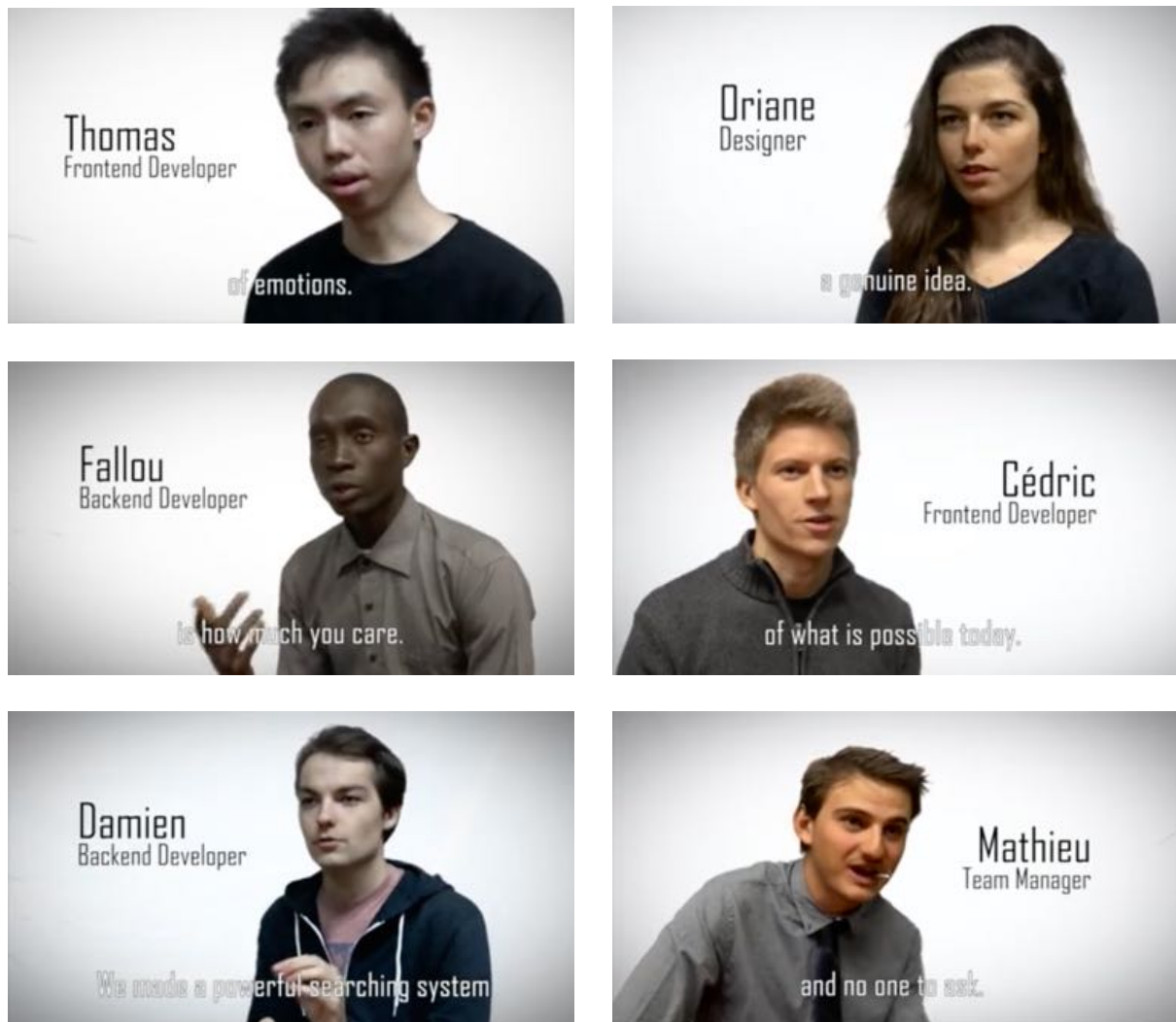


FIGURE 5.2 – Quelques captures de la vidéo proposée

Publiée sur la page Facebook officielle du Challenge (7600 personnes atteintes), elle permet de faire décoller les téléchargements (4330 vues de la vidéo).

## 5.2 La communication interne à l'École

### 5.2.1 Communication pour les VP Tournois

Puisque chaque VP Tournoi doit utiliser l'application pour entrer les scores du tournoi auquel il est responsable, une communication en amont de l'événement a dû être effectuée pour les former. Même si la manipulation est très simple, une rapide formation a été effectuée lors d'une AG de Formation des VP Tournois et des documents leur ont été fournis (Annexe A, Figures A.1 et A.2).

Des identifiants de connexion leur ont été distribués, ainsi que leur mot de passe associé.



### 5.2.2 Communication avec l'équipe du Challenge

La communication avec l'équipe du Challenge a commencé très tôt. Rapidement, nous avons pris contact avec différents membres de l'équipe. Beaucoup d'informations techniques sur l'organisation du tournoi nous sont parvenues.

Toutefois, certaines n'ont pu être communiquées qu'au dernier moment. L'ensemble des matchs qui se sont joués pendant le Challenge ont été communiqués à notre équipe 72h avant le début de l'événement. Sachant qu'il fallait en moyenne 1h de saisie par match et par genre, la saisie de l'ensemble des données (pour remplir la base de données) a pris en moyenne 10h par personne de l'équipe.

Par ailleurs, l'application présente un système de Newsfeed : des News sont publiées (sur la page d'accueil) régulièrement au cours du week-end pour prévenir les participants des moments forts. Cette liste de News (Annexe B) a été donnée à notre équipe quelques jours avant le week-end. Dès lors, des membres de l'équipe se sont chargés de remplir la base de données en temps réel avec ces informations aux heures indiquées par le Challenge.

### 5.2.3 Communication avec le PE 90 Challenge durable

La communication avec l'équipe du PE Challenge Durable se résume essentiellement à l'implémentation de leur travail dans l'application. Les deux équipes ont commencé à travailler ensemble tardivement. Ce n'est qu'à partir de Décembre que les deux équipes avaient une vision claire de leur projet.

Le PE Challenge Durable (Figure 5.3) a dans un premier temps réalisé son travail seul puis est venu intégrer leur travail dans l'application. La communication consistait alors à leur imposer le design de leur contenu avec aussi quelques restrictions quant à la forme qu'il pouvait avoir tout en leur laissant la possibilité de publier l'intégralité du travail qu'ils avaient fait.



FIGURE 5.3 – Logo du Challenge durable

## Partie 6

# Après le Challenge : résultats et analyses

### 6.1 Téléchargements

Notre objectif étant de développer une application, nous souhaitons que celle-ci soit téléchargée un maximum de fois. Le nombre de téléchargements uniques est proposé Figure 6.1 :

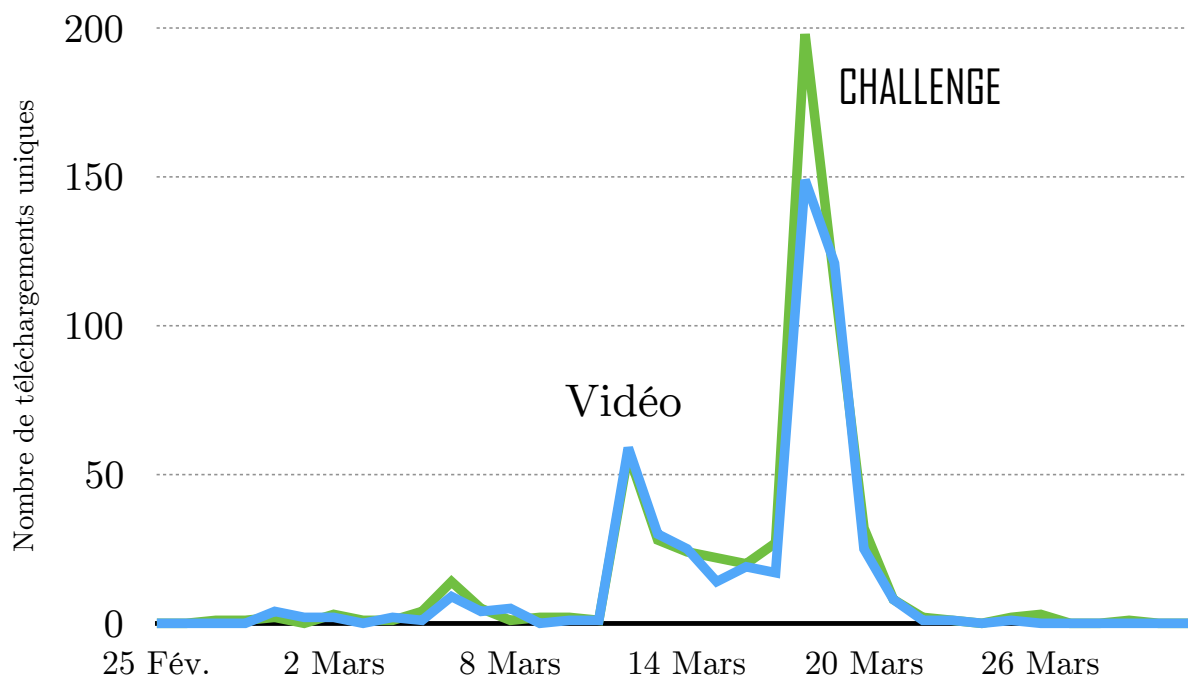


FIGURE 6.1 – Graphique montrant le nombre de téléchargements de l’application (en bleu pour iOS et en vert pour Android)



La répartition des téléchargements selon la plateforme utilisée est donnée ci-dessous (Figure 6.2) :

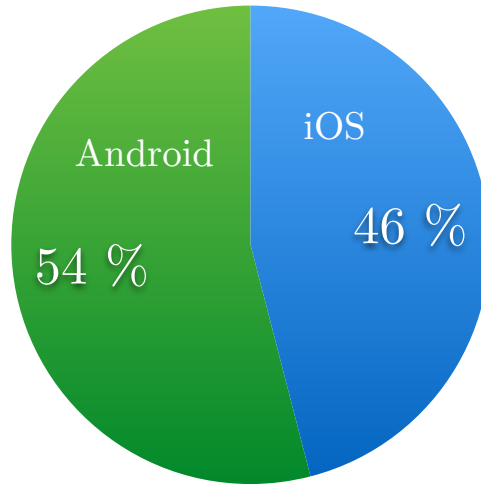


FIGURE 6.2 – Graphique montrant la répartition des téléchargements de l'application (en bleu pour iOS et en vert pour Android)

Aussi, des données concernant les appareils qui utilisent l'application ont été récupérées (Figure 6.3). Les smartphones sont dominants mais quelques tablettes l'utilisent aussi (nous pensons que les téléchargements effectués sur ordinateurs sous plateforme iOS sont dûs à Apple qui a effectué des tests).

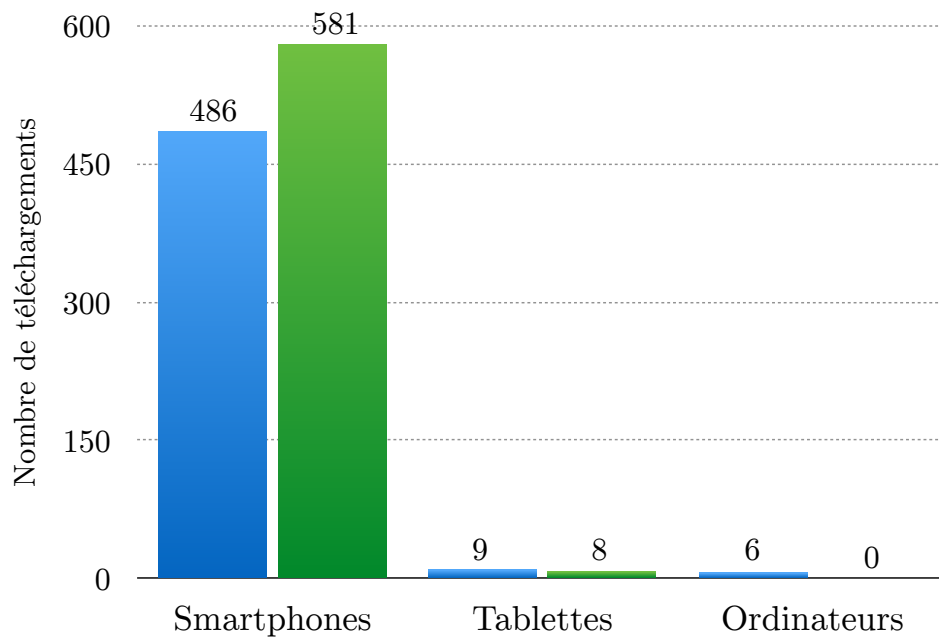


FIGURE 6.3 – Graphique montrant la répartition des téléchargements de l'application selon l'appareil utilisé (en bleu pour iOS et en vert pour Android)

Sur environ 3000 participants, on comptabilise 1090 téléchargements : un peu plus d'un tiers des participants.

Avec un peu de recul, nous avons appris que la quasi-totalité des Centraliens de Lyon n'avaient pas téléchargé l'application. L'émulation sur le campus et la communication directe qui précédaient l'événement leur avaient permis de connaître bien en amont le déroulement du week-end et ses spécificités. Par conséquent, ils n'ont pas jugé utile de la télécharger directement.

iOS	Android	TOTAL
501 (46 %)	589 (54 %)	1090

TABLE 6.1 – Récapitulatif des téléchargements

De même, des données concernant le territoire où se situaient les utilisateurs ont été collectées. La plupart se situe en Europe (en France) mais quelques utilisateurs se trouvaient aussi en Asie, en Amérique et aux Caraïbes et en Afrique. Certains élèves de l'École, en stage à l'étranger, ont certainement suivi l'événement de loin... !

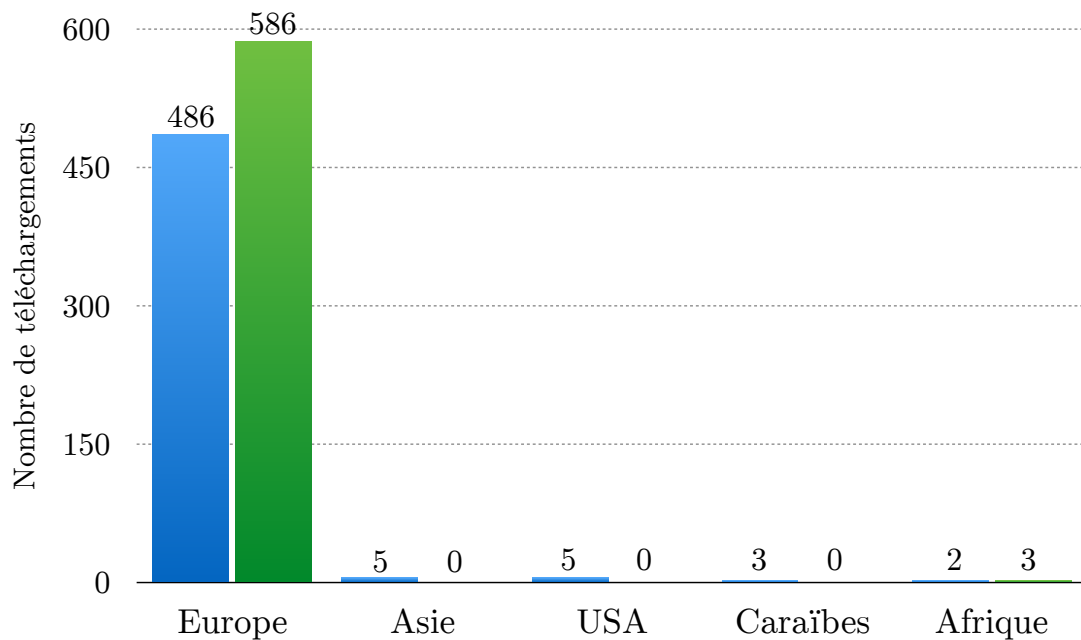


FIGURE 6.4 – Graphique montrant la répartition des téléchargements de l'application selon le territoire (en bleu pour iOS et en vert pour Android)

## 6.2 Retours des utilisateurs

Suite à la soumission de l'application sur l'AppleStore et le PlayStore, quelques retours concernant l'application ont été soumis. Ils sont listés ci-dessous :

*Indispensable ! Cette appli contient toutes les infos dont on a besoin pour vivre le Challenge à fond. Grande réussite et bravo aux développeurs.*

*Aymeric Moulene*

*Merci les developpeurs. Elle tient toutes ses promesses.*

*Baptiste Fabre-Perrin*

*Ca brasse beaucoup trop au PE 29 ! Une bombe.*

*thoto970325*

*C'est bien mais... Il faut mettre les résultats pour tous les sports maintenant !*

*Quentin Remy*

*Bon travail app fantastique, ne peut pas attendre pour l'événement !*

*Donatien Delehelle*

*Super travail ! Blblbl*

*Bastien Gauthier*

*Bonne appli, demande a mûrir pour être optimum.*

*Emmanuel GUEHO*

*Génial ! Cette application est vraiment super ! On y trouve toutes les informations nécessaires pour s'éclater au Challenge ! À télécharger !*

*Quentin GUEHO*

*Super ! Vraiment pratique. J'adore.*

*Roland Lamidieu*

Certains commentaires participent au folklore autour de l'application. Les remarques publiées sont très appréciées de l'équipe car elles sont une forme de récompense du travail fourni. Toutefois, il est aussi intéressant d'avoir des critiques et des remarques concernant les choses qui ne fonctionnent pas, qui pourraient être améliorées ou encore qui n'ont pas du tout eu l'effet attendu.

Pour en savoir plus et afin d'obtenir des critiques plus précises, nous avons soumis un questionnaire (Annexe B) que certains participants et organisateurs ont rempli.

## 6.3 Analyse des retours

Parmi l'ensemble des retours du questionnaire (Annexe B), certaines remarques redondantes ont permis d'identifier des problèmes.

### 6.3.1 Galerie photo

Beaucoup des utilisateurs ont trouvé que la galerie photo apportait une réelle plus value pour l'application. Néanmoins, un certain nombre n'ont pas trouvé l'accès à cette galerie (taper sur une des photos de la page d'accueil).

#### *Analyse*

*Peut-être que l'accès à la Galerie n'est pas très intuitif. Il faut effectivement taper sur une des photos présentes en page d'accueil pour y accéder. Une solution serait d'indiquer l'accès à la Galerie par une indication légère et subtile. Pourquoi pas intégrer un onglet en bas de page.*

### 6.3.2 La présentation des matchs

Si on s'intresse à la présentation des matchs, on retrouve trois grands axes de plainte :

- les matchs ne sont pas assez visibles : les séparations entre deux matchs différents ne sont pas assez prononcées ;
- les couleurs des différentes phases du tournoi sont trop sobres et ne mettent pas en avant la programmation ;
- l'affichage des poules n'est pas assez satisfaisant : les utilisateurs auraient aimé une sous-section pour chaque poule (Figure 6.5, page suivante).

#### *Analyse*

*Les séparations ne sont effectivement pas très marquées pour les phases de poules. Les matchs de poules sont listés les uns à la suite des autres, ce qui donne une lecture linéaire et pas ergonomique. En revanche, les démarcations sont suffisamment prononcées pour les phases finales.*

*Ce choix a été décidé suite à un manque de compétence technique. L'idée originelle était d'afficher les poules sous forme d'un tableau à double entrée. Chaque case aurait alors été constituée des informations du match en question.*

*D'une manière générale, il faut augmenter la visibilité de la page Matches, à la fois pour l'affichage des poules et des matchs en général. Il faut travailler au niveau du design.*

*Malgré tout, l'affichage des matchs est un véritable défi. Il faut afficher une quantité conséquente d'informations, d'une manière claire et concise, tout en respectant les contraintes imposées comme le (très) peu de place à l'écran. Et puisque l'application peut être installée sur tous types de téléphones - et donc toutes tailles d'écran - elle doit s'adapter au matériel de l'utilisateur.*

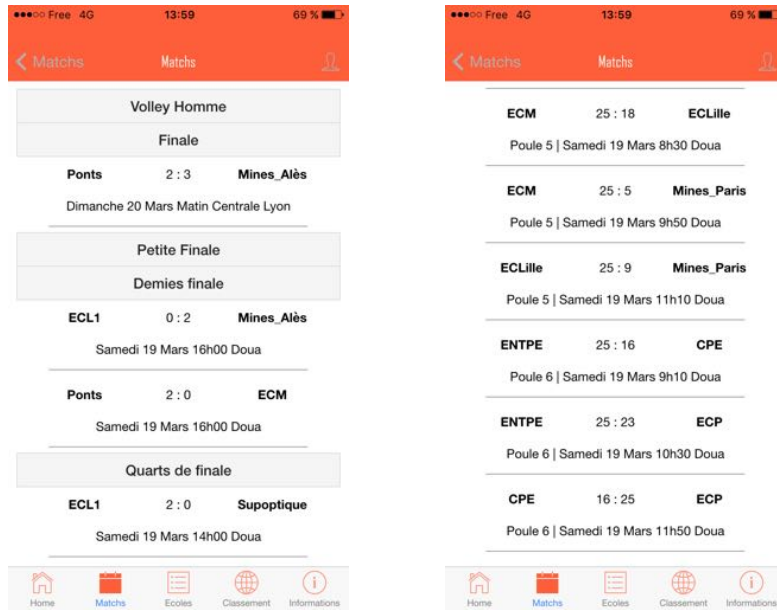


FIGURE 6.5 – Affichage du tournoi de volley masculin avec phases finales et phases de poule

### 6.3.3 L'entrée des sports par les VP Tournois

Les VP Tournois avaient un rôle primordial dans la réussite de l'application au cours du week-end. Ils étaient en effet chargés de remplir au fur et à mesure les résultats des matchs de leur sport. La rapidité d'entrée des résultats dans l'application était un facteur déterminant. Le relevé est indiqué Figure 6.6 :

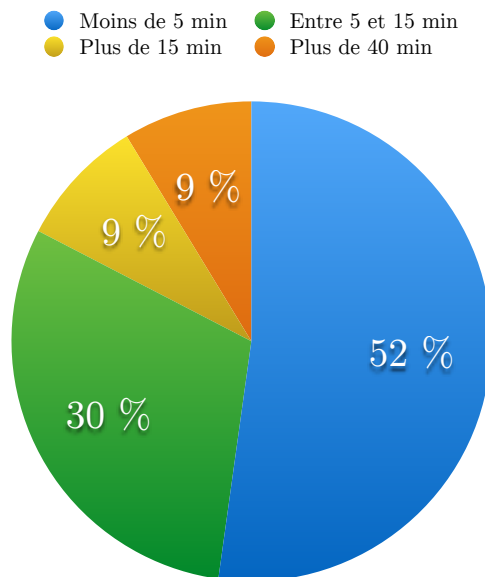


FIGURE 6.6 – Temps de remplissage moyen des résultats des matchs dans l'application par les VP Tournois

Selon les VP Tournois, l'application a été utilisée de plusieurs manières :

- soit ils ne l'ont pas utilisée du tournoi à cause d'un problème directement lié à l'application :
  - l'application n'était pas disponible sur son portable (WindowsPhone) ;
  - il y avait des problèmes d'affichage à l'écran pour l'interface connectée des VP Tournois (Figure 4.5) (problème de calibrage) ce qui rendait impossible l'entrée des scores et des matchs ;
  - manque de forfait internet.
- soit ils l'ont utilisée seulement à la fin du tournoi (pour les sports individuels par exemple, où seul un classement final est effectué) ;
- soit ils n'ont pas pu l'utiliser à cause du format du tournoi :
  - changement du format de certains sports en cours de tournoi et les informations fournies dans l'application n'étaient plus compatibles avec ce qui se passait en réalité ;
  - par manque de temps.

Certains ont tout de même parfaitement réussi !

### **Analyse**

*Les VP Tournois ont certainement été les utilisateurs les plus actifs sur l'application et entraient les informations sans doute les plus importantes et les plus attendues.*

#### Problèmes directement liés à l'application

*Nous n'avons pas étudié le cas des téléphones Windows puisque cela représente un très petit pourcentage des smartphones en circulation (< 6 %). Le choix était délibéré pour économiser du temps de codage puisqu'ionic ne prenait pas en compte le système d'exploitation Windows sur Windows Phone.*

*Les problèmes d'affichages à l'écran ont été repérés mais pas pu être résolus à temps. La seule solution était de taper exactement sur le score afin de le modifier, et pas à côté.*

*Le manque de forfait internet était un risque à prendre de notre part, sachant que les forfaits sont de plus en plus conséquents et que la couverture en 3G/4G est de plus en plus performante. Cela a touché très peu de VP Tournois (3 personnes).*

#### Problèmes liés au format du tournoi

*En cours de tournoi, si la programmation changeait brusquement (comme en Handball masculin par exemple où une deuxième phase de poule s'est vue rajoutée au dernier moment), les VP Tournois n'avaient aucun moyen de la modifier. Pour rappel, il faut se connecter au site <http://challenge-2016.eclair.ec-lyon.fr/Formulaire/index.php> (web-app de gestion de notre base de données challenge2016) pour modifier la programmation des matchs. Impossible à faire pour un VP Tournoi.*

*Aussi, certains VP Tournois étaient littéralement débordés. Le tournoi qu'ils avaient à gérer leur demandait toute leur attention, leur énergie et même plus, ce qui a éclipsé l'application.*

## Partie 7

# Bilan et perspectives futures

### 7.1 Bilan de notre travail

Notre équipe s'est attachée au respect du cahier lors du développement de l'application. La majeure partie des fonctionnalités présentes dans le cahier des charges ont pu être mises en place. En effet, voici un bilan de ce qui a été réalisé.

En ce qui concerne la prise en main de l'application, une application native, téléchargeable sur les deux plateformes pour applications les plus importantes, a été développée. La fonction principale que propose l'application est la mise à disposition de diverses informations concernant l'événement sportif. Pour ceci, l'application devait être construite de la manière la plus ergonomique possible (avec un nombre de clic minimal pour accéder à une information donnée par exemple) et devait avoir une structure claire et intuitive. D'après les retours collectés à l'aide du questionnaire, les objectifs d'ergonomie et de lisibilité de l'application ont pu être atteints. L'affichage de tant d'information peut nécessiter un certain temps, car il faut toutes les récupérer auprès d'un serveur stockant ces données. Ceci représente un des points qui nécessitent d'être retravaillés. En effet, le temps de traitement et d'affichage des données dynamiques est quelque fois un peu long. Toutefois, la partie statique de l'application est plutôt réactive, en effet, lorsqu'on clique sur un bouton, l'affichage de la nouvelle page est quasi-instantané.

L'information et le contenu en général devaient aussi couvrir l'ensemble de l'événement, donc avoir accès aux photos prises durant le challenge, à des informations générales, aux scores des matchs etc.. Pour réaliser ceci, une galerie photo a été mise en place, couplée à une slideshow permettant de prévisualiser trois photos contenues dans cette galerie. Cette galerie est accessible à travers la page d'accueil en cliquant sur une image du slideshow. Afin que les personnes participant au tournoi soient au courant des dernières actualités, un newsfeed, lui aussi présent sur la page d'accueil, a été implémenté. Ce newsfeed contient les petites annonces postées par l'équipe du challenge à travers le site qui a été développé. De plus, s'agissant d'un tournoi, l'accès aux scores des matchs et aux différentes données relatives aux sports proposés était un facteur essentiel que l'application devait être capable de proposer. Pour ceci a été mis en place une page classement, affichant le classement des écoles en temps réel, une page match permettant d'afficher les matchs d'un certain sport et une page école permettant d'afficher tous les matchs d'une école. Pour

que les participants puissent s'organiser, tous les matchs programmés sont affichés avec la date, l'heure et le lieu en complément. L'affichage des scores est dynamique, les VP Tournois les rentrent sur la page Matchs et ils se mettent automatiquement à jour pour les autres utilisateurs.

De plus, un onglet spécial Informations a été mis en place contenant une FAQ, un plan et autres informations que pourraient rechercher les participants.

Un des points faisant le charme de l'application est son contenu dynamique. Effectivement, tout un système d'affichage d'information en temps réel a été mis en place. Les contenus dynamiques sont : la galerie photo, le Newsfeed, le classement et les scores des matchs. Les pages statiques ne constituent en fait qu'une faible partie de l'application. Afin d'assurer ce dynamisme, une base de donnée contenant toutes les informations mentionnées plus haut a été créée. Cette base de donnée est couplée à un API permettant la communication de l'application et du site avec celle-ci.

Quelles sont alors les différences avec ce qui a été produit par l'équipe de l'année dernière ?

## 7.2 Comparaison

Les caractéristiques de l'application proposée par l'équipe précédente ont été explicitées plus haut, comparons désormais les deux applications. La majeure différence entre les deux applications est le fait que celle de l'année dernière n'est pas native (pas développée dans les langages spécifiques aux deux systèmes d'exploitation iOS et Android) contrairement à celle qui a été développée par notre équipe, mais était une web-app, donc accessible à partir d'un « site » internet. On observe donc que la plupart des fonctionnalités présentes dans l'application précédente le sont aussi dans la nouvelle application.

Seuls l'affichage des partenaires du challenge et des informations concernant les navettes ne sont plus présentent dans la nouvelle application. Il n'a pas été jugé nécessaire d'afficher les informations concernant les navettes car l'expérience de l'année précédente a démontré que les personnes ne respectaient pas nécessairement les horaires et géraient leurs déplacements de manière autonome.

Des fonctionnalités ont été ajoutées dans cette nouvelle version de l'application et n'étaient pas présentent dans celle d'avant. Il s'agit de la galerie photo, du classement en temps réel, de la possibilité de se connecter pour rentrer les scores, le formulaire, la page développement durable, le zoom sur les images. De plus, un grand travail sur l'ergonomie de l'application a été effectué, le but était de réduire un maximum le nombre de clics nécessaires pour accéder à une informations. Ceci nous paraît être une grande plus-value comparée à l'application précédente, qui nécessitait généralement beaucoup de clics pour accéder à une information.



Fonctionnalités	Web-app 2015	Challenge ECL 2016
Application native	❌	✅
Affichage de News	✅	✅
Galerie photos	❌	✅
Zoom sur les images	❌	✅
Programmation des tournois	✅	✅
Score des matchs	✅	✅
Connexion pour entrer les scores	❌	✅
Classement en temps réel	❌	✅
Programme du week-end	✅	✅
Plans	✅	✅
FAQ	✅	✅
Numéros d'urgence	✅	✅
Page de développement durable	❌	✅
Partenaires du Challenge	✅	❌
Informations concernant les navettes	✅	❌
Site de gestion de base de données	❌	✅
Réalisation d'un manuel technique	❌	✅
Notifications	❌	❌

TABLE 7.1 – Comparaison de l'application 2016 Challenge ECL avec la web-app de 2015

Pour la bonne compréhension du fonctionnement de l'application et surtout du code qui a été écrit, un manuel technique (Annexe G) a été rédigé afin d'expliquer tout ceci. Ceci a été un des points qui ont été jugés important pour ceux qui utiliseront et essaieront d'améliorer l'application par la suite. En effet, certains problèmes de bonne compréhension du fonctionnement de l'application de l'an dernier ont été rencontrés, ce qui a rendu sa réutilisation impossible. Il a donc semblé essentiel de faire en sorte que la prise en main de l'application et de son code soit aussi aisée que possible, afin que les suivants aient une vraie base sur laquelle s'appuyer.

L'équipe précédente avait proposé un serveur sms afin de pouvoir entrer les scores. Nous avons choisi une méthode différente, qui a été de pouvoir entrer les scores directement sur l'application, car elle nous a paru être la solution la plus naturelle. En effet, pour le bon fonctionnement d'un serveur sms, il est nécessaire de respecter une certaine syntaxe

dans le sms envoyé. Cela représente une grande contrainte pour les VP Tournois qui n'ont pas toujours le temps de s'occuper de ceci lorsqu'ils sont dans le feu de l'action. Il a aussi été estimé que le fait de devoir se connecter à internet et le manque de batterie n'était pas un problème si important que ça, compte-tenu du fait que les connexions internet mobiles sont aujourd'hui globalement plutôt bonnes et que la longévité des batteries sont suffisantes pour une utilisation intensive durant une journée.

Une fonctionnalité que l'on souhaitait ajouter à la nouvelle version était l'envoi de notifications, qui n'a pas pu être mise en place dû à une priorisation des fonctionnalités à intégrer et à un temps, disponible pour le développement, limité. Les notifications n'ont pas été jugées indispensables dans le bon fonctionnement de l'application.

## 7.3 Qu'en est-il pour la suite ?

L'analyse des retours a permis de mettre en évidence quelques points qui restent à améliorer et qui justifient le renouvellement du PE.

Voici donc l'esquisse d'un nouveau cahier des charges :

- l'accès à la galerie photo doit être plus intuitif, dans la nouvelle application on pourra songer à ajouter un élément plus visuel indiquant l'accès à la galerie, comme un onglet en bas de page par exemple ;
- la présentation actuelle des matchs n'étant pas satisfaisante car trop peu lisible, la nouvelle équipe s'attachera à l'importance de la clarté de l'affichage des matchs, en améliorant la séparation entre les différents matchs et les différentes poules, et en cherchant à plus mettre en avant la programmation, à travers un choix stratégique des couleurs par exemple ;
- proposer le déploiement de l'application sur WindowsPhone en complément du déploiement sur iOS et Android afin qu'un maximum de personnes puisse utiliser l'application ;
- améliorer le système d'entrée de données afin qu'il n'y ait plus de problèmes liés à l'affichage ;
- améliorer aussi le système afin que l'entrée soit plus rapide et plus efficace afin que les VP Tournois perdent un minimum de temps lorsqu'ils entrent les résultats ;
- prévoir plus en amont les changements de dernières minutes de programmation dans certains sports (comme ça a été le cas pour le Handball cette année).

# Conclusion

Le principal objectif du PE 29 : Application pour le Challenge était de créer une application native, agréable à utiliser et intuitive du point de vue de l'utilisateur, avec un contenu dynamique, afin de proposer un véritable outil de communication pour l'équipe du Challenge. Pour réaliser l'application souhaitée, un cahier des charges précis a été développé, puis une étude théorique de sa structure a été réalisée (combien de pages, relations entre celles-ci, structure de la base de données etc..) et finalement le codage en lui-même a été effectué.

Le cahier des charges fixé initialement a pu globalement être respecté, hormis quelques points tel que le système de notifications qui n'a pas pu être implémenté par manque de temps. Le fonctionnement de l'application finale a été satisfaisant, surtout à la vue des retours que l'on a eu et à la vue du nombre de téléchargements (plus de 1000), ce qui montre que l'application a su apporter une véritable plus value au déroulement du tournoi. De plus, l'application s'est avérée être un moyen de communication puissant, et l'équipe du Challenge a pu en profiter, en envoyant 38 news tout au long du week-end.

Des problèmes ont été soulevés lors des retours par les VP Tournois et des utilisateurs, problèmes qui sont donc autant de points que l'on peut encore améliorer. Une compilation de ces soucis a été réalisée, et certaines pistes de solution ont été proposées, afin que nos successeurs puissent se lancer directement dans la bonne voie.

Enfin, dans l'idée que l'équipe qui reprendra le flambeau en Septembre puisse démarrer son travail dans des conditions optimales, un manuel technique (Annexe G) explicitant le fonctionnement du code développé a été rédigé.

# Annexe A

## Communication pour les VP Tournois



### À l'attention des VP Tournoi...

Cette année, le Challenge dispose d'une application qui permet à tous les participants d'avoir une vue d'ensemble sur l'événement. Entre autres, les VP Tournoi disposent d'un accès privilégié pour entrer les scores ainsi que la programmation des phases finales des sports collectifs.

Pour les sports individuels, les VP Tournoi pourront entrer le classement final.

Rendez-vous tous sur l'App Store ou sur le Play store pour télécharger la dernière version (version 1.1) :



AppStore



Challenge ECL



PlayStore

Vous recevrez vos comptes pour vous connecter Mercredi 16 Mars en soirée ou Jeudi 17 Mars dans la journée.

Il y a un compte par sport : plusieurs VP Tournoi peuvent donc être connectés sur le même compte.

Nous comptons sur vous pour mettre à jour le plus régulièrement possible les scores et la programmation des phases finales. Dans le cas où vous n'êtes pas connecté à Internet, envoyez ce qui est nécessaire aux deux numéros suivants :

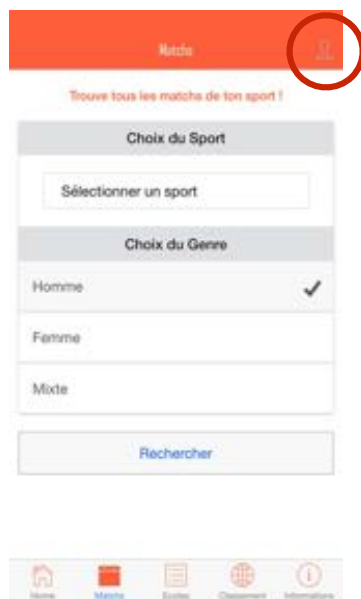
Damien Guého (Blum) : 06 62 47 48 30

Serigne Fallou Ndiaye : 06 18 76 54 42

FIGURE A.1 – Première page de la documentation distribuée aux VP Tournois

### Attention

Le Challenge et notre équipe du PE vous faisons confiance.  
Lorsque vous êtes connecté, vous pouvez modifier tous les scores de tous les matchs de toutes les phases ainsi que la programmation des phases finales.  
Nous vous demandons donc de mettre à jour uniquement la partie vous concernant.  
Surtout, déconnectez-vous lorsque vous n'avez plus besoin d'utiliser l'application !



### Pour se connecter

- Cliquez sur l'icône en forme de tête en haut à droite (présent sur toutes les pages) et entrez votre login et votre mot de passe ;
- Allez sur la page match (page affichée ici à gauche) cherchez le sport concerné et sélectionnez le genre ;
- Vous pourrez ensuite indiquer les écoles et les scores des phases finales et de poules pour les sports collectifs et le classement des sports individuels ;
- Déconnectez-vous ;)

FIGURE A.2 – Deuxième page de la documentation distribuée aux VP Tournois

Ces documents ont été fournis environ une semaine avant le Challenge à une soixantaine de VP Tournois.

## Annexe B

### Questionnaire de satisfaction pour les utilisateurs de l'application



**Retour sur l'application du Challenge**

L'application vous a-t-elle servie d'une quelconque manière pendant le Challenge ?

☐ Oui

☐ Non

Avez-vous remarqué les news ajoutées durant le week-end ?

☐ Oui

☐ Non

Avez-vous apprécié la présence d'une galerie photo ?

☐ Oui

☐ Non

FIGURE B.1 – Première partie du questionnaire

Qu'avez-vous pensé du design global de l'application ?

- ☐ Excellent
- ☐ Plutôt pas mal
- ☐ Pas très bon
- ☐ Vilain

L'application vous a-t-elle parue intuitive, facile d'utilisation ?

- ☐ Oui
- ☐ Ça dépend
- ☐ Non

D'une manière générale, avez-vous eu facilement accès aux informations que vous recherchiez ?

- ☐ J'ai facilement eu accès aux actualités du week-end : news et galerie photos
- ☐ J'ai facilement eu accès aux matchs que je recherchais
- ☐ J'ai facilement eu accès aux informations concernant une école
- ☐ J'ai facilement eu accès au classement complet
- ☐ J'ai facilement eu accès aux réponses à mes questions

FIGURE B.2 – Deuxième partie du questionnaire

Que pensez-vous de la présentation des différents matchs (pour le Basketball, Handball, Football, Rugby et Volleyball) ?

- ☐ C'est très bien
- ☐ La disposition est à améliorer
- ☐ Illisible !

Question pour les VP Tournois : la saisie des scores était-elle assez efficace ?

- ☐ Oui
- ☐ Assez mais avec quelques problèmes de programmation tout de même
- ☐ Non, pas du tout

Avez-vous des remarques / améliorations / questions / réclamations à faire à l'équipe ?

Votre réponse

---

FIGURE B.3 – Troisième partie du questionnaire



# Annexe C

## Liste des News

### C.1 Vendredi

20h : Le coin chill est ouvert ! Vous êtes arrivés et les sièges du bus vous ont laissé dans un sale état ? Venez vous poser et passer une soirée tranquille.

21h : C'est l'heure de la soirée casino ! Rendez-vous dès maintenant dans le foyer de Centrale pour se retrouver autour d'une table de poker et passer une bonne soirée.

22h : Ouverture des activités sous le chapiteau. Au programme, balayette infernale et Just Dance Now. On vous attend nombreux !

### C.2 Samedi

6h15 : Le RU ouvre ses portes pour vous nourrir avant cette journée haute en rebondissements !

8h : Le RU ferme dans 30 min, dépêchez-vous !

11h : Le coin chill est ouvert. Si vous êtes encore sur le campus, passez faire un tour !

12h : Le village Challenge est ouvert ! Des entreprises et des associations sont venues spécialement pour vous présenter leurs projets et vous proposer de nombreuses activités !

12h : De nombreuses activités se tiennent sur le campus : combat de sumos (zone 1), pronostiques sportifs (zone 2), cirque (zone 4).

18h : Les fanfares des différentes écoles se retrouvent sous le chapiteau pour mettre l'ambiance !

18h : De retour sur le campus avec des courbatures partout ? Allez vous faire masser par les ostéos au W1bis.

18h : Les activités reprennent ! Rappel : combat de sumos (zone 1), pronostiques sportifs (zone 2), cirque (zone 4).

18h30 : C'est l'heure du dîner ! Le RU ouvre ses portes pour votre plus grand plaisir.

19h : C'est l'heure de la finale de Hand masculin au gymnase ! Allez soutenir votre équipe préférée !

20h : Après le Hand, c'est maintenant l'heure des finales de Badminton, Tennis de table et escalade au gymnase. Grand spectacle en perspective !

20h30 : La soirée commence ! Ne loupez pas le JT dans 30 minutes ! Récap de la journée, interview des supporters. Il ne faut manquer ça pour rien au monde !

21h15 : Ne loupez pas le SHOW POMPOM dans 15 min au chapiteau ! Les équipes spécialement sélectionnées pour le Challenge vont se disputer le titre de meilleures pompoms !

21h30 : Le RU ferme dans 30 min, dépêchez-vous !

## C.3 Dimanche

00h15 : Vous êtes ambiancés comme jamais ? Ne loupez pas Androma dans 15 min au foyer !

00h45 : La première navette part dans 15 min. Il y en a toute les demi-heure jusqu'à 5h30.

2h15 : Après une super prestation, Androma laisse la place à Omnia dans 15 min pour ambiancer les planches du foyer !

3h30 : Rappel : des navettes pour Bellecour sont en place jusqu'à 5h30.

5h15 : Dernière navette vers Bellecour dans 15 min !

7h : Le RU est ouvert. Le tournoi n'est pas fini, il te faut des forces pour donner le meilleur de toi-même !

8h : Maintenant, tout de suite, c'est le début de la finale de Basket féminin au gymnase. Viens encourager ton équipe préférée !

8h30 : C'est l'heure de la finale de foot masculin sur le stade ! Ambiance de folie garantie !

9h : Les ostéos sont de retour pour votre plus grand plaisir et vous propose des massages dans le W1bis.

9h : Après les dames, c'est au tour de ces messieurs de se disputer la première place au basket, maintenant dans le gymnase !

9h : Le RU ferme dans 30min, dépêchez-vous !

9h : Les activités cirques continuent en zone 4 !

9h30 : C'est l'heure des finales de Foot féminin et Rugby féminin maintenant sur le stade !

10h : Le coin chill est ouvert, venez vous détendre !

10h30 : C'est l'heure de la course de mascotte ! RDV sur le stade maintenant pour venir profiter du spectacle !

10h30 : C'est l'heure de la finale de Hand féminin dans le gymnase ! On vous attend nombreux !

11h : Cette année, le Challenge met le handisport à l'honneur ! Allez vous initier au Handitennis sur les courts, démonstration et activités seront au rendez-vous !

11h : Le choc des titans ! C'est l'heure de la finale de Rugby masculin sur le stade. Ça va chauffer !

11h30 : Allez vous initier au Cécifoot au gymnase. Une équipe de la région vient spécialement sur le campus pour vous présenter ce sport hors du commun.

11h45 : Le RU est ouvert ! C'est déjà le dernier repas à Centrale...

13h30 : Le RU ferme dans 30 min, dépêchez-vous !

# Annexe D

## Coûts

### D.1 Les coûts financiers

Les dépenses financières sont très rapide à commenter. Nous avons acheté deux comptes développeurs (un compte Apple Developer au nom de Damien Guého, un compte Google Developer au nom de Serigne Fallou Ndiaye). Quelques euros ont été dépensés en impression au SDeC (Service Des Cours de Centrale).

Les dépenses sont rassemblées dans le Tableau D.1 :

Dépense	Montant
Compte développeur Apple	99,00 €
Compte développeur Google	25,00 €
Impressions	20,00 €
<b>TOTAL</b>	<b>144,00 €</b>

TABLE D.1 – Récapitulatif des dépenses

### D.2 Les coûts en temps

De Septembre à Juin nous travaillions environ 6h par semaine. Début Février, pendant 2 semaines, juste avant la première validation, chaque jour (et nuit !) était l'occasion d'avancer le développement de l'application. Nous estimons 4h de travail par jour environ. De mi-Février jusqu'au week-end du Challenge, nous estimons à minimum 5h par semaine le temps de travail (débugage, corrections, nouvelle soumission). Enfin, pendant le week-end, trois heures ont suffi.

Après le Challenge, le traitement des retours et les corrections mineures de l'application ont demandé une dizaine d'heures de travail.

Donc :

$$\begin{aligned}\text{Nombre d'heures travaillées} &= 6\text{h par semaine} \times 16 \text{ semaines} \\ &+ 4 \text{ heures par jours} \times 15 \text{ jours} \\ &+ 5 \text{ heures par semaines} \times 5 \text{ semaines} \\ &+ 3 \text{ heures} \\ &+ 10 \text{ heures}\end{aligned}$$

Soit un total de 194 heures par personne, ou encore presque 1000 heures de travail pour le groupe.

# Annexe E

## Présentation de l'équipe

### E.1 Membres de l'équipe

Notre équipe se compose de 6 membres :



Cédric ERNEWEIN



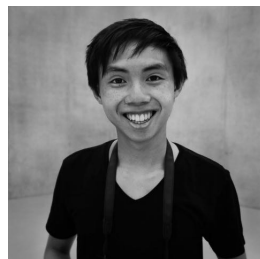
Mathieu GAULTIER



Damien GUÉHO



Oriane GUYON



Thomas KUOCH



Serigne Fallou NDIAYE

### E.2 Répartition des tâches

Au sein d'un projet tel que celui-ci, la répartition des tâches ne fût pas évidente. En effet, tout le projet consiste en la création d'une application smartphone. Par conséquent, le projet se compose essentiellement d'un programme informatique ce qui restreint les possibilités dans la répartition des tâches. Un programme informatique forme un tout et chaque morceau du programme à son impact sur les autres parties de ce dernier.

Une fois l'architecture de l'application décidée et l'ensemble des pages créées avec les liens entre chacune d'elles, il était alors possible de travailler indépendamment sur chacune des pages. Dès lors, chaque membre de l'équipe a pu se focaliser sur des pages spécifiques.

La répartition des tâches s'est alors fait de la manière suivante :

- Damien, chef de projet, est responsable de s'assurer de l'avancée de l'application afin qu'elle soit terminée à temps. Il est également le chargé de la communication avec l'équipe du Challenge (commanditaire du PE). D'un point de vue programmation, il est chargé d'une partie de la programmation frontend et du backend ;
- Fallou est codeur frontend et backend et responsable de l'hébergement de l'application chez ÉCLAIR. Du fait de ses compétences, il supervise l'avancée du développement ;
- Cédric est un des principaux codeurs frontend et responsable en partie de la communication sur Facebook ;
- Mathieu est codeur frontend et est en partie responsable de communications extérieures ;
- Thomas est codeur frontend et est responsable de l'intégration du PE « Challenge Durable » dans l'application ;
- Oriane, étudiante en architecture, donc moins présente sur le projet a cependant été consultante en design sur l'application.

## Annexe F

### Diagrammes de Gant

Les diagrammes de Gant nous ont aidé à ne surtout pas oublier la date limite du début Challenge le 18 Mars, et nous ont permis de s'organiser en amont sur la répartition des tâches.

Les Figures F.1 et F.2 montrent l'organisation temporelle de notre travail. Les plannings ont été assez respectés avec néanmoins 2 à 3 jours de retard à partir du mois de Janvier et une semaine d'avance à partir du mois de Mars.

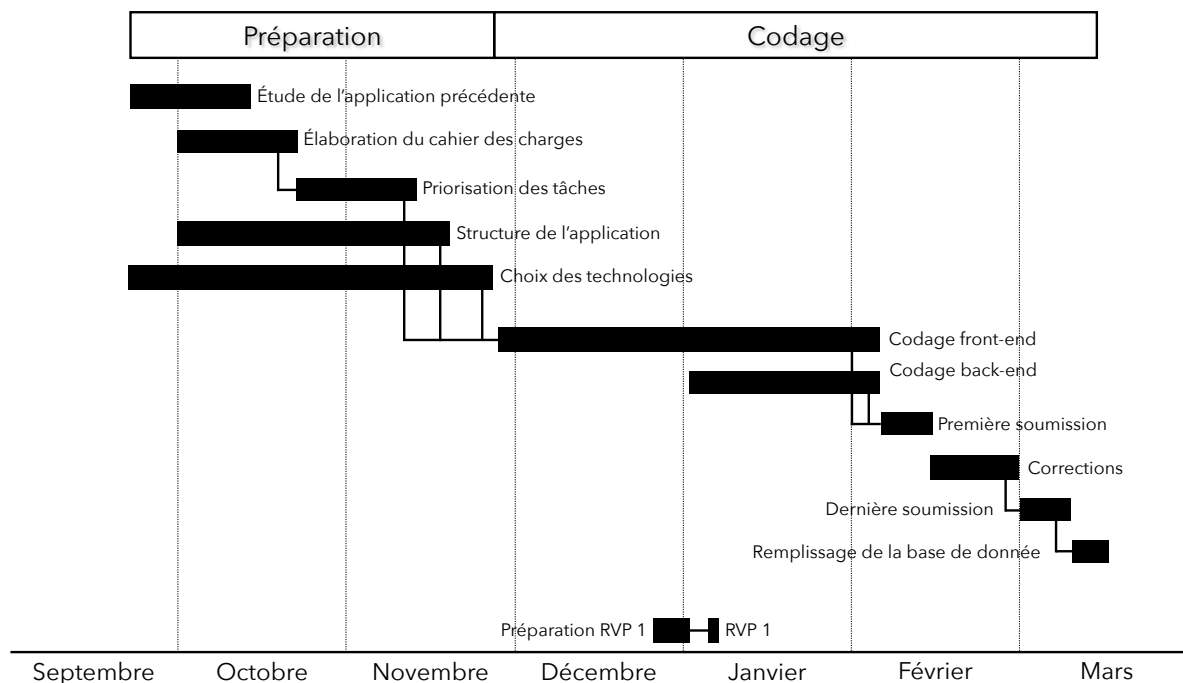


FIGURE F.1 – Organisation de la première période



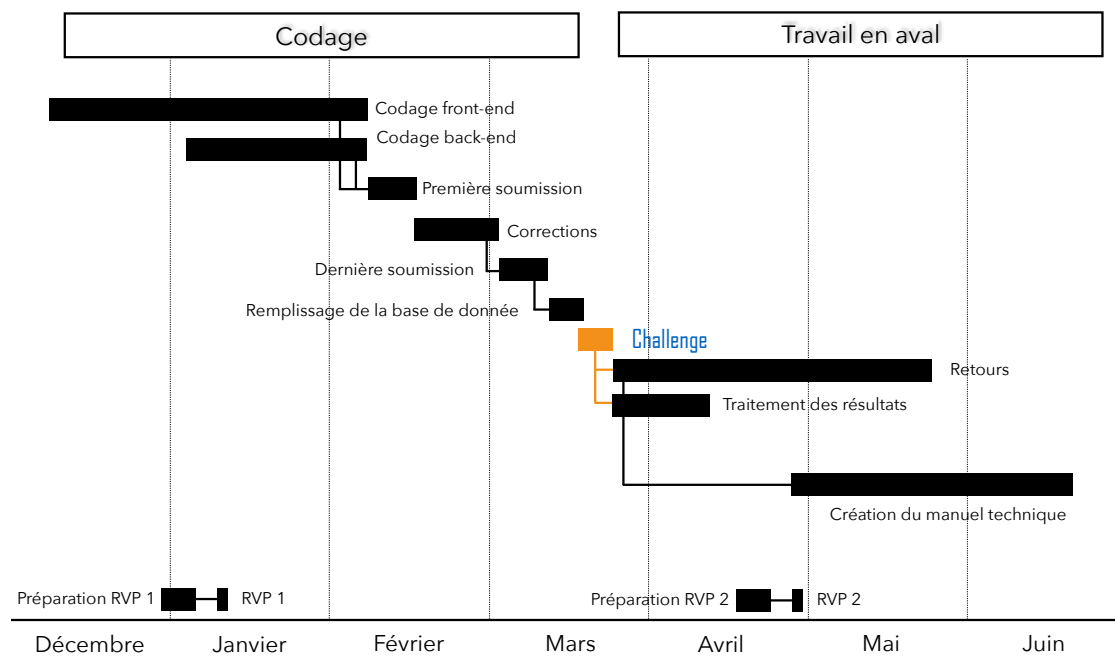


FIGURE F.2 – Organisation de la deuxième période

# Annexe G

## Manuel technique

### Préliminaire

La bonne compréhension du manuel technique de l'application Challenge ECL requiert des connaissances basiques en langage HTML/CSS, PHP et Javascript (voire même des bases dans l'utilisation du framework AngularJs, framework utilisé par ionic), ce manuel ne peut remplacer un cours de ces langages informatiques. Ce manuel technique est destiné à un public averti.

### G.1 Développement du Frontend

#### G.1.1 Installation de Ionic

Pour pouvoir développer le frontend, il nous faut installer ionic. Ionic produira ensuite un projet de base avec un certain nombre de fichiers.

Pour installer Ionic, il faut d'abord installer les plateformes Javascript Node.js et npm.

Voici la méthode à suivre, à partir de la console de commande `cmd` :

```
$ sudo apt-get install nodejs
$ sudo apt-get install npm
$ npm install -g cordova ionic
$ ionic start challenge tabs
$ cd challenge
$ ionic serve
```

#### G.1.2 Structure du dossier de travail

Un dossier nommé `challenge` est créé dans le dossier de l'utilisateur (Figure G.1). On y retrouve les dossiers `hooks`, `platforms`, `plugins`, `scss` et `www`.

L'essentiel du codage se fera dans le dossier `www` (Figure G.1). Dans celui-ci, il y a les dossiers `css`, `img`, `js`, `lib`, `templates` et un fichier `index.html`.

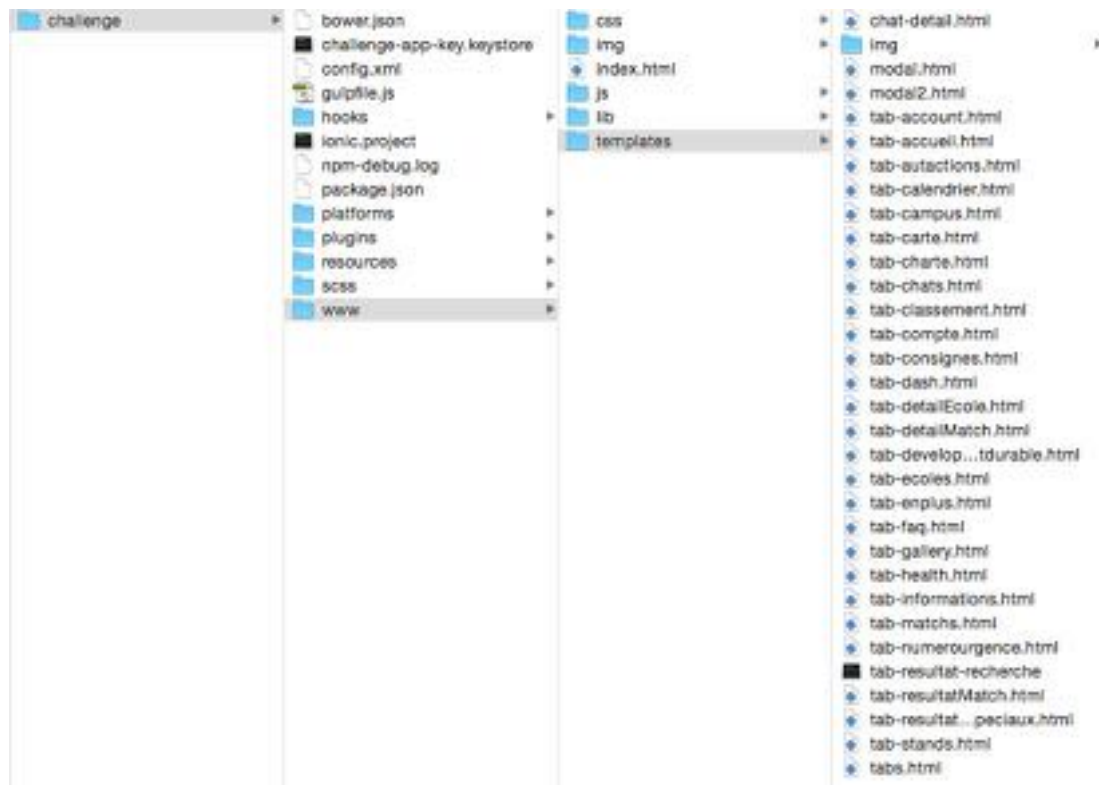


FIGURE G.1 – Dossier de travail de l’application

Le dossier `www` contient :

- un dossier `css` contiendra une feuille de style `.css` pour la présentation de l’application : couleurs, polices, tailles... ;
- un dossier `img` contiendra les images qui apparaissent lors de la navigation dans l’application ;
- un dossier `js` contiendra trois fichiers javascript, `app`, `controllers` et `services` qui assurent le lien entre les différentes pages de l’application ;
- un dossier `lib` qui contient les modules nécessaires au fonctionnement du framework ;
- un dossier `templates` contient toutes les pages de l’application. Chaque page doit avoir le nom suivant : “`tab-nom`”. Par exemple, la page relative aux informations porte le nom “`tab-informations`”. Par contre cette nomenclature des pages n’est pas une condition obligatoire pour le fonctionnement de l’appli mais elle est très conseillée par les développeurs Ionic car elle permet une certaine cohérence du code.

Le codage du frontend se fera essentiellement dans les dossiers `js` et `templates`. Les images utilisées seront placées dans le dossier `img`, et la feuille de style dans le dossier `css` est modifié pour obtenir la présentation souhaitée.

### G.1.3 Pré-visualiser l'application sur l'ordinateur

Pour visualiser le projet en cours, c'est-à-dire l'application mobile qui est en train d'être développée, comme sur un smartphone, il faut retourner dans la console de commande et taper les lignes de codes suivantes (en sachant que notre projet a pour nom **challenge**) :

```
$ cd challenge  
$ ionic serve
```

Une fenêtre navigateur s'ouvrira dans laquelle l'application mobile sera accessible de manière locale (**localhost**). C'est ainsi que l'on pourra visualiser et tester les modifications apportées à l'application lors du codage. Pour avoir une vue "mobile" (Figure G.2), c'est-à-dire avec des résolutions d'écrans propres à un smartphone, on effectue, sur le navigateur la combinaison **Ctrl + Shift + M**.

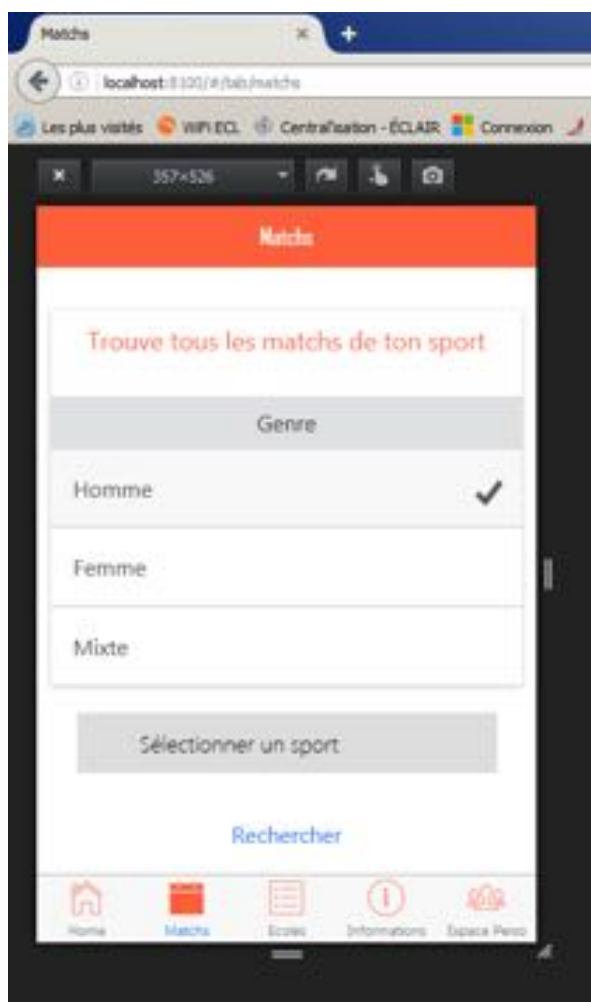


FIGURE G.2 – Pré-visualisation de l'application mobile sur ordinateur - Page Matches

## G.1.4 Codage des différentes pages de l'application

### La barre de navigation (tabs)

L'application est divisée en cinq parties, accessibles à partir de la barre de navigation (Figure G.3). Les cinq pages majeures sont :

- La page d'accueil (Home) ;
- la page Matches ;
- la page Ecoles ;
- la page Classement ;
- la page Informations.



FIGURE G.3 – Barre de navigation de l'application

Un accès à un espace administrateur est également disponible, à condition de posséder un identifiant administrateur et le mot de passe.

La barre de navigation est située en haut pour la version Android, en bas pour la version iOS.

Le paramétrage de la barre de navigation se fait grâce au fichier html `tabs` située dans le dossier `templates`.

Pour définir la barre de navigation, on utilise la balise :

```
<ion-tabs class="tabs-icon-top tabs-color-active-positive">
</ion-tabs>
```

Pour définir un onglet de la barre de navigation, on utilise le code :

```
<ion-tab title="Home" icon-off="ion-ios-home-outline custom-icon"
  icon-on="ion-ios-home custom-icon" href="#/tab/accueil">
<ion-nav-view name="tab-accueil"></ion-nav-view>
</ion-tab>
```

Dans ce code, qui définit l'onglet "Home" :

- `title="Accueil"` définit le nom de l'onglet qui apparaîtra dans la barre de navigation
- `icon-off` et `icon-on` permettent de définir les icônes qui apparaissent lorsque l'on se trouve (on), ou non (off) sur l'onglet. La valeur qui suit fait référence à des icônes prédéfinis par Ionic.
- `href="#/tab/accueil` renvoie à la page vers laquelle on souhaite naviguer (en l'occurrence, vers la page "tab-accueil")
- `name="tab-accueil"` fait référence à la page "tab-accueil".

## Code général d'une page - Exemple de la page "Informations"

Les différentes pages de l'application, à l'exception de la page Home, sont structurées à l'aide de blocs content un titre et une description. La Figure G.4 est une capture d'écran de la page Informations.



FIGURE G.4 – Page Informations

Le codage "classique" d'une page de l'application est décrit dans ce paragraphe. C'est ainsi que la page Informations a été codée.

Le code de la page est encadrée par les balises suivantes :

```
<ion-view view-title="Titre">
<ion-content class="padding">
...
</ion-content>
</ion-view>
```

Le terme `view-title="Titre"` dans la première ligne définit le titre de la page consultée, dans l'application. Par exemple, si l'on a `view-title="Infos"`, alors on retrouve la Figure G.4. Pour obtenir des blocs, par exemple le bloc "Développement durable" de la Figure G.4, on écrit le code suivant :

```
<div class="list card">
<a href="#/tab/developpementdurable">
<div class="item item-divider custom-item">Développement durable</div>
<div class="item item-body">
Toutes les infos pour unchallenge durable ! </br>
</div>
</a>
</div>
```

Dans ce code :

- la balise contenant la classe “`list card`” permet de définir le bloc ;
- la balise de lien `<a href="#tab/page"> ... </a>` permet de naviguer vers une nouvelle page, en l’occurrence la page “`tab-page`”, répertoriée dans le dossier `templates` ;
- la balise content la classe “`item item-divider custom-item`” définit le titre du bloc ;
- la balise contenant “`item item-body`” définit le texte dans le bloc.

On remarquera que la balise `<a> ... </a>` englobe les balises content les classes “`item item-divider custom-item`” et “`item item-body`”. Cela permet à l’utilisateur d’accéder à la page référencée en touchant le bloc entier sur son écran (et non une ligne de texte).

## Naviguer d’une page à l’autre

On a vu que la balise `<a> ... </a>` permet de faire le lien entre deux pages. Dans le code-exemple précédent, on s’attend donc à ce que lorsque l’on clique (ou touche) le bloc “Développement durable”, on accède à la page `tab-developpementdurable`.

Il reste cependant à apporter des modifications dans les fichiers `controllers` et `services` dans le dossier `js` pour que cela fonctionne. En reprenant l’exemple du bloc “Développement durable”, la démarche à adopter est décrite ci-dessous.

Dans le fichier `controllers`, écrire le code :

```
.controller('DeveloppementdurableCtrl', function($scope) {  
  })
```

Dans le fichier `app`, écrire le code :

```
.state('tab.developpementdurable', {  
  cache: true,  
  url: '/developpementdurable',  
  views: {  
    'tab-informations': {  
      templateUrl: 'templates/tab-developpementdurable.html',  
      controller: 'DeveloppementdurableCtrl'  
    }  
  }  
})
```

Dans ce code, tous les termes “`developpementdurable`” font référence à la page `tab-developpementdurable` dans le dossier `templates`.

La logique du code impose que le point de départ vers cette page se situe dans à l’onglet-page `tab-informations`.

La ligne `controller: 'DeveloppementdurableCtrl'` fait directement référence au `controller` que l’on a défini précédemment dans la page `controllers`.

Le clic, qui permet de passer de la page “Informations” à la page “Développement durable”, en touchant le bloc, est maintenant effectif.

Si l’on veut, à partir du bloc "Développement durable", naviguer vers un sous-bloc à l’intérieur, par exemple un bloc "consignes", la méthode reste, si ce n’est que la page `tab-consignes` reste liée à la page `tab-informations`, et non à ce que l’on pourrait croire à la page `tab-developpementdurable`. Dans le fichier `controllers`, écrire le code :

```
.controller('ConsignesCtrl', function($scope) {  
  })
```

Dans le fichier `app`, écrire le code :

```
.state('tab.consignes', {  
  url: '/consignes',  
  views: {  
    'tab-informations': {  
      templateUrl: 'templates/tab-consignes.html',  
      controller: 'ConsignesCtrl'  
    }  
  }  
})
```

## Page classement

La page classement (Figure G.5) comporte un bloc "Classement Général". On y retrouve le classement des écoles participantes au Challenge du premier au dernier, les logos, les noms et le nombre de points obtenus.










Classement Général		
1	 Ecole Centrale Lyon	12840 pts
2	 Ecole Centrale Nantes	8820 pts
3	 Agro ParisTech	5120 pts
4	 Ecole Polytechnique fédérale de Lausanne	4180 pts
5	 Ecole Centrale Lille	4020 pts
6	 Mines d'Alès	3260 pts
7	 ISAE	3000 pts
8	 ESTP	2800 pts
9	 Vieux cons	2580 pts

FIGURE G.5 – Page Classement



La page "Classement" est gérée par le controller suivant, écrite dans le fichier `controllers` dans le dossier `js` :

```
.controller('ClassementCtrl', ['$scope', '$http', function ($scope, $http) {

$http.get('http://challenge-2016.eclair.ec-lyon.fr/api/ecoles/classement').
  then(function(resp) {
    $scope.classement = resp.data.data;

  }, function(err) {
    console.error('ERR', err);

  });
}]);
```

Dans ce code :

- Une requête html **GET** au niveau de l'API **RESTful** permet de récupérer le classement. On stocke ce classement dans la variable `\$scope.classement` faisant le lien avec la Vue (`\$scope` étant la vue)
- Si jamais il devait y avoir une erreur, on envoie le statut d'erreur correspondant.

Le but de cette page est, comme son nom l'indique, d'afficher le classement général des écoles.

Pour cela, on a les balises usuelles utilisées par Ionic. On a ainsi le code suivant pour la page "Classement" :

```
<div class="item item-divider"><div class="titreinfo"> Classement
  Général</div></div>
<div class="row">

<div class="col"><b></b></div>
<div class="col"><b>Rang</b></div>
<div class="col"><b>Ecole</b></div>
<div class="col"><b>Points</b></div>
</div>
<div ng-repeat="ecole in classement">
<div class="row">

<div class="col"></div>
<div class="col">{{$index +1}}</div>
<div class="col">{{ecole.nom_ecole}}</div>
<div class="col">{{ecole.points}}</div>
</div>
</div>
```

Dans ce code :

- On construit un tableau, toutes les lignes sont composées de quatre colonnes.
- La première contient un élément vide dans la première colonne, les mots **Rang**, **Ecole** et **Points** respectivement dans les deuxième, troisième et quatrième colonnes
- La première colonne sera utilisée afin d'afficher le logo de l'école
- AngularJs permet ensuite d'itérer sur les lignes et les éléments de la variable `$scope.classement` mentionnée précédemment.
- Dans chaque ligne on va afficher le logo de l'école, son rang, pour ceci on incrémente la variable `{{index}}` à chaque ligne, son nom et son nombre de points.

## Page d'accueil "Home"

La page d'accueil (Figure G.6), notée "Home" dans la barre de navigation, s'articule en deux parties : une galerie photos et un bloc des dernières informations, appelées "news".

Trois images de la galerie sont chargées et défilent. Lorsqu'on clique sur une des photos, on accède alors à la galerie.

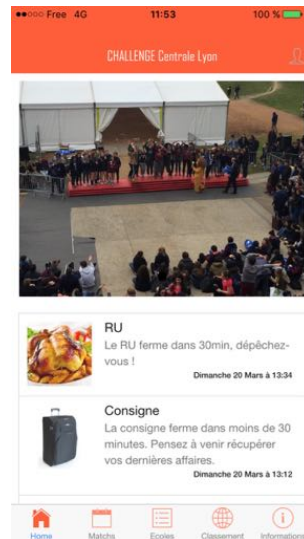


FIGURE G.6 – Page Home

La page "Home" est gérée par le controller suivant, dont le code dans le fichier `controllers` du dossier `js` est :

```
.controller('AccueilCtrl', ['$scope', '$ionicHistory', '$ionicSlideBoxDelegate',
'$http', '$localStorage', '$timeout', 'Home'],
function($scope, $ionicHistory, $ionicSlideBoxDelegate, $http, $localStorage,
$timeout, Home) {

    $scope.navSlide = function(index) {
        $ionicSlideBoxDelegate.slide(index, 500);
    }

    Home.Lien().then(function (resp) {
```

```

$scope.liens = resp.data;

}, function (err) {
console.error('ERR', err);
});

Home.News().then(function (resp) {
var infos = resp.data;
$scope.news=infos;

}, function (err) {
console.error('ERR', err);
});
}])

```

Dans ce code :

- Dans une première partie on peut observer la fonction associée à la slideshow visible sur la page d'accueil;
- La deuxième partie du code permet de récupérer les images de la galerie que l'on voudra afficher. Ces images sont obtenues à l'aide du service `Home` (voir plus loin);
- On stocke ces images dans la variable `$scope.liens`.

Dans le fichier html `tab-accueil` du dossier `templates`, le code pour coder la galerie est :

```

<ion-slide-box pager-click="navSlide(index)" auto-play=
  "true" does-continue="true">
<ion-slide>
<a href="#/tab/gallery">

</a>

</ion-slide>
<ion-slide>
<a href="#/tab/gallery">

</a>
</ion-slide>
<ion-slide>
<a href="#/tab/gallery">

</a>
</ion-slide>
</ion-slide-box>

```

Dans ce code :

- La slideshow de la galerie est entourée par les balises :  
`<ion-slide-box>`  
`</ion-slide-box>`  
Ceci correspond à un environnement de slideshow proposé par Ionic.
- Chaque élément de la slideshow est entouré des balises :  
`<ion-slide>`  
`</ion-slide>`
- Chaque élément de la slideshow est une image de la galerie qui est de plus cliquable grâce aux balises `<a> ... </a>`
- La source des images est contenue dans la variable `liens`

Ensuite, la deuxième partie, la partie “news” quant à elle est gérée par les parties suivantes :

```
Home.News().then(function (resp) {
var infos = resp.data;
$scope.news=infos;
//$localStorage.setObject('news',infos);

}, function (err) {
console.error('ERR', err);
});
//$timeout($scope.news = $localStorage.getObject('news'),1);

})
```

Dans ce code, on récupère, comme auparavant pour les images, les news à l’aide du service `Home` et on stocke ces news dans la variable `$scope.news`

Dans le fichier `services`, le controller relié au service `home` est le code :

```
.factory('Home',function($http, $q){

return{
News:function(){
var deferred = $q.defer();
$http({
method : 'GET',
url : 'http://challenge-2016.eclair.ec-lyon.fr/api/home/infos',
//params :{id:1}

}).success(function (response){
// This is the data from your request
deferred.resolve(response)
}).error(function () {
// If something was wrong
```

```

deferred.reject();
});
return deferred.promise;
},
Gallery:function(){
var deferred = $q.defer();
$http({
method  : 'GET',
url     : 'http://challenge-2016.eclair.ec-lyon.fr/api/home/galerie',
//params :{id:1}

}).success(function (response){
// This is the data from your request
deferred.resolve(response)
}).error(function () {
// If something was wrong
deferred.reject();
});
return deferred.promise;
},
Lien:function(){
var deferred = $q.defer();
$http({
method  : 'GET',
url     : 'http://challenge-2016.eclair.ec-lyon.fr/api/home/galerie/liens',
//params :{id:1}

}).success(function (response){
// This is the data from your request
deferred.resolve(response)
}).error(function () {
// If something was wrong
deferred.reject();
});
return deferred.promise;
}
}
})

```

- Il y a trois grandes parties : les fonctions qui permettent de récupérer respectivement les données pour les news, la galerie et les images de la slidebox ;
- Les trois fonctionnent de la même manière : On effectue une requête `GET` auprès de l'API, qui va renvoyer les données que l'on aura demandées sous forme JSON ;
- Si c'est un succès, on stocke dans la variable correspondante, sinon on envoie un code d'erreur.

```
<div class="list">

<div ng-repeat="item in news" class="item item-thumbnail-left item-text-wrap" >



<h2>{{ item.titre_info }}</h2>
<p>{{ item.contenu_info }}</p>
<h6 align="right" class="subdued"><i>{{item.date_info}} à
    {{item.heure_info}}<i></h6>
</div>
</div>
```

- On crée une liste
- AngularJs va s'occuper d'itérer sur la variable news puis va afficher chaque donnée contenue dans la variable, c'est-à-dire `item.image_info` correspondant à l'image associée au news que l'on a envoyée, `item.titre_info` correspondant au titre de la news, `item.contenu_info` correspondant au contenu de la news et `item.date_info` correspondant à la date et l'heure à laquelle la news a été envoyée

```
.controller('MatchesCtrl', ['$scope', '$http', '$state', '$ionicPopup', 'Matches', function

$scope.recherche = function(matches){
var sport=matches.sport;
var genre=matches.sexe;
//Resultat de la recherche
if(sport){
Matches.Matches(sport,genre).then(function (resp) {
// Here will be the data after the request finished
var res = resp.data;
if(res.sport==="Football" || res.sport==="Basketball" || res.sport==="Handball" || re
```

```

//$localStorage.setObject('classement',donnees);
$state.go("tab.resultatMatch", { resultat: res});
}
else{
$state.go("tab.resultatMatchSpeciaux", { resultat: res});
}

}, function (err) {
// Here will be if there was an error
console.error('ERR', err);
});
}
else{
$scope.alertPopup = $ionicPopup.alert({
title: 'Quel Sport?',
template: 'Veuillez choisir un sport!'
});
}
}

$scope.sexe=[
{text:'Homme', value:'Homme',logo:'img/homme.png'},
{text:'Femme', value:'Femme',logo:'img/femme.png'},
{text:'Mixte', value:'Mixte',logo:'img/mixte.png'}
];

$scope.matches = {
sexe: 'Homme',
};

$scope.sports=[
{text:"Basketball", valeur:'Basketball', logo:'/img/Logos_sport/basket.png'},
{text:"Football", valeur:'Football', logo:'/img/Logos_sport/foot2.png'},
{text:"Handball", valeur:'Handball', logo:'/img/Logos_sport/handball.png'},
{text:"Rugby", valeur:'Rugby', logo:'/img/Logos_sport/rugby.png'},
{text:"Volley", valeur:'Volley', logo:'/img/Logos_sport/volley.png'},
{text:"Athlétisme", valeur:'Athletisme', logo:'/img/Logos_sport/athle.png'},
{text:"Badminton", valeur:'Badminton', logo:'/img/Logos_sport/badminton.png'},
{text:"Escalade", valeur:'Escalade', logo:'/img/Logos_sport/escalade.png'},
{text:"Escrime", valeur:'Escrime', logo:'/img/Logos_sport/escrime.png'},
{text:"Golf", valeur:'Golf', logo:'/img/Logos_sport/golf.png'},
{text:"Judo", valeur:'Judo', logo:'/img/Logos_sport/judo.png'},
{text:"Natation", valeur:'Natation', logo:'/img/Logos_sport/natation.png'},
{text:"Raid", valeur:'Raid', logo:'/img/Logos_sport/raid.png'},
{text:"Ski", valeur:'Ski', logo:'/img/Logos_sport/ski.png'},

```

```

{text:"Tennis", valeur:'Tennis', logo:'/img/Logos_sport/tennis.png'},
{text:"Tennis de table", valeur:'Tennis_table', logo:'/img/Logos_sport/tennistable.png'},
{text:"Ultimate", valeur:'Ultimate', logo:'/img/Logos_sport/ultimate.png'},
{text:"Waterpolo", valeur:'Waterpolo', logo:'/img/Logos_sport/waterpolo.png'},
];

})

```

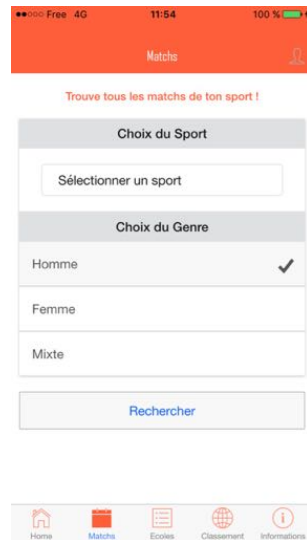


FIGURE G.7 – Page Matches

Dans ce code :

- On définit la fonction recherche, qui va permettre de rechercher les matchs en fonction du sport et du sexe sélectionnés. La fonction prend en argument une variable `match` contenant les informations sur le sexe et le sport sélectionné. On distingue les cas de sport individuel ou collectif et fonction de ceci on charge soit la page `tab.resultatMatch` pour les sports collectif, et `tab.resultatMatchSpeciaux` pour les sports individuels.
- Il y a trois variables associées au `$scope` : `$scope.sexe` qui contient les valeurs féminin, masculin et mixte , `$scope.matches` qui permet d'avoir une valeur par défaut pour les boutons radio fixée sur masculin et `$scope.sport` qui contient tous les sports présents au Challenge.

Le code du template `tab-matches` est le suivant :

```

<h5 class="phrase_description">Trouve tous les matchs de ton sport !</h5>
<div class="list card">
<div>
<div class="item item-divider custom-item mini_titre" style="text-align:center"><di
<br>
<div class="bouton_selection">

```



```

<select class="ion-input-select" name="singleSelect" ng-model="matches.sport">
<option value="">Sélectionner un sport</option> <!-- not selected / blank option -->
<option ng-repeat="sport in sports" ng-value="sport.text"> {{sport.text}} </option>
</select>
</div>
<br>
<div class="item item-divider custom-item mini_titre" style="text-align:center"><div>
<!--<input type="radio" name="sexe" value="Homme" placeholder="" ng-model="matches.sport">
<ion-radio class = "custom-radio-sexe" ng-repeat="item in sexe" ng-value="item.value">
{{item.text}}
</ion-radio>
</div>
</div>

<div class="button-match">
<button class="button button-block button-stable custom-button" type="submit" ng-click="search(match)">Rechercher
</div>

```

Dans ce code, il y a trois parties :

- La première est un **select** pour sélectionner le sport souhaité. Ce select est construit à l'aide d'un **ng-repeat** qui itère sur la variable **sports** dans laquelle sont stockés tous les sports ;
- La deuxième partie correspond à des boutons radio qui permettent de sélectionner le sexe. Donc féminin, masculin ou mixte. Ces boutons radio sont à nouveau construits à l'aide d'un **ng-repeat** ;
- La troisième partie du code est le bouton de recherche qui lorsqu'on clique dessus va exécuter la fonction **recherche**. On lui donne en paramètre la variable **match**, dans laquelle on a deux attributs : **match.sport** et **match.sexe** qui ont été déterminés grâce aux deux parties précédentes.

## Page Ecoles

La page "Ecoles" (Figure G.8) permet à l'utilisateur d'afficher tous les matchs d'une école choisie.

Le contrôleur qui gère la page Ecoles a pour code :

```

.controller('EcolesCtrl', ['$scope', '$http', '$localStorage', '$timeout', 'Ecole', function($scope, $http, $localStorage, $timeout, Ecole) {

//Listes des Ecoles
Ecole.Ecoles().then(function (resp) {
// Here will be the data after the request finished
var donnees = resp.data;
$scope.ecoles=donnees;
//$localStorage.setObject('ecoles',donnees);

}, function (err) {
// Here will be if there was an error
console.error('ERR', err);
});

```

```

});
//$timeout($scope.ecoles = $localStorage.getObject('ecoles'),1);

$scope.getInfoEcole= function(Infoecole){
Ecole.Matches(Infoecole).then(function (resp) {
$scope.matches_ecole = resp.data;
}, function (err) {
// Here will be if there was an error
console.error('ERR', err);
})
}
}
}]]

```



FIGURE G.8 – Page Ecoles

Dans ce code, il y a trois parties :

- La première partie du code permet de récupérer la liste des écoles présentes pour le challenge. Cette liste a été récupérée grâce au service '**Ecole**', qui va effectuer une requête **GET** à l'API pour obtenir les informations souhaitées (comme déjà expliqué plus haut) ;
- La deuxième partie du code est une fonction qui est appelée lorsqu'on appuie sur le bouton de recherche présent sur la page ;
- A partir de l'école sélectionnée dans le **select**, elle va récupérer toutes les informations de l'école à nouveau à l'aide du service '**Ecole**' et de la fonction **Matches** qu'elle contient qui va effectuer une requête **GET** pour récupérer tous les matchs correspondant à l'école choisie.

Le template de la page est le suivant :

```
<h5 class="phrase_description">Trouve tous les matchs de ton école !</h4>

<div class="list card titreinfo">
<div class="item item-divider custom-item mini_titre"><div class="titreinfo">Informat
<div class="item item-body">
<form name="myForm">

<select class="ion-input-select" name="singleSelect" ng-model="Infoecole">
<option value="">Sélectionner une école</option> <!-- not selected / blank option -->
<option ng-repeat="ecole in ecoles | orderBy: 'nom_ecole'" value="{{ecole.abreviation
</select>
<br>
<br>
<a><button class="button button-block button-stable custom-button" ng-click="getInfoE

</div>
</form>
</div>
</div>
<div ng-if="matchs_ecole">
<div class="row">
<div class="col col-25"></div>
<div class="col col-50"><b>{{matchs_ecole.nom_ecole}}</b></div>
<div class="col col-25"><b>{{matchs_ecole.points}} pts</b></div>
</div>
<div class="item item-divider"><div class="titreinfo">Matchs à venir</div></div>

<ul>
<li ng-repeat='match in matchs_ecole.matchs | filter: {"etat_match":"0"}'>
<div class="row"><div class="col">{{match.phase_match}}</div></div>
<div class="row">
<div class="col">{{match.equipe1}}</div>
<div class="col">{{match.equipe2}}</div>

</div>
<div>{{match.date_match}} {{match.heure_match}} {{match.lieu_match}}</div>
<HR class="barre" size=1 width="80%">

</li>
</ul>

<div class="item item-divider"><div class="titreinfo">Matchs joués</div></div>

<ul>
```

```

<li ng-repeat='match in matchs_ecole.matches | filter: {"etat_match":"1"}'>
<div class="row"><div class="col">{{match.phase_match}}</div></div>
<div class="row">
<div class="col">{{match.equipe1}}</div>
<div class="col">{{match.score1}} : {{match.score2}}</div>
<div class="col">{{match.equipe2}}</div>
</div>
<HR class="barre" size=1 width="80%">
</li>
</ul>

</div>
</div>

```

Dans ce code :

- La première partie du code correspond à la construction d'un `select` comme cela a été le cas pour la page `matches` ;
- Dans une deuxième partie, on va afficher toutes les informations concernant l'école, donc tout d'abord son nom et son nombre de points ;
- On affiche alors les matchs qui n'ont pas encore été joués, à l'aide d'un `ng-repeat` itérant sur la variable `match_ecole.matches`, en filtrant à partir de toute la liste des `matches`, les `matches` qui ont leur état égal à "0", ce qui correspond à un match non encore joué. On affiche alors toutes les informations concernant chaque match ;
- On fait de même pour les `matches` qui ont déjà été joués, mais qui ont cette fois-ci leur état fixé à "1".

A titre d'exemple, voici ce que l'on obtient (Figure G.9) lorsqu'on sélectionne "Ecole Centrale de Lyon"

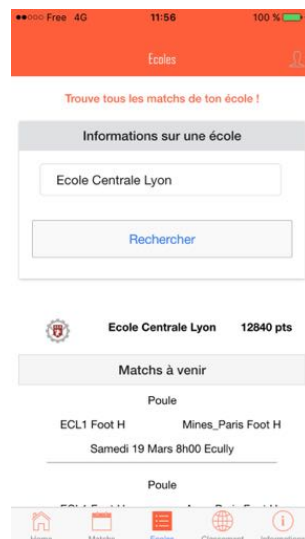


FIGURE G.9 – Recherche des matchs de l'Ecole Centrale de Lyon

### G.1.5 Le CSS

La feuille de style `style.css` contenu dans le dossier `css`, lui-même contenu dans `www` permet de coder la mise en forme, la présentation de l'application : couleurs, polices, marges, bordures, positionnements ...

Dans cette feuille de style, on retrouve le codage de la mise en forme des balises qui ont servi à encadrer les pages de l'application.

Ainsi, la barre de navigation est mise en forme à l'aide du code suivant :

```
.custom-bar{
border-color: rgb(209, 121, 58);
background-color: rgb(209, 121, 58);
color: #fff;
font-family: 'Agency FB';
```

On retrouve le terme `custom-bar` qui fait référence à la balise `<custom-bar>`, encadrant le template `tabs`, que Ionic associe à la barre de navigation.

Pour respecter la charte graphique du Challenge, la police *Agency FB* a été adoptée pour les titres des blocs, balisés `titreinfo`, et *Century Gothic* pour les textes intérieurs, balisés `corpsinfo`. On retrouve ces informations dans le code ci-dessous :

```
.titreinfo {
font-family: "Agency FB";
font-size: x-large;
}

.corpsinfo {
font-family: "Century Gothic";
}
```

## G.2 Déploiement de l'application

Avant de détailler la procédure, voici un tutoriel qui explique bien comment se fait la publication pour Android et iOS :

<http://ionicframework.com/docs/guide/publishing.html>.

### G.2.1 Compilation sous Android

La compilation est le processus qui va permettre de transformer le code AngularJs (html, css, js ...) en native et de générer l'application avec une extension `.apk` . Cette phase de compilation est réalisée par Cordova qui se trouve déjà intégrée dans Ionic.

Lors de la compilation, il faudra signer l'application avec une clé (Keystore) pour que celle-ci soit identifiée de façon unique dans le PlayStore. Par ailleurs, après chaque modification du code (mise à jour de l'application), il faut nécessairement recompilée en

utilisant cette clés. Sinon la mise à jour de l'application dans le PlayStore ne sera pas acceptée. Si la clé se perd, il sera impossible de mettre à jour l'application.

La clé utilisée lors de la compilation de l'application a été générée par nous-même sous Linux.

Pour plus de détails sur cette procédure, voici un tutoriel intéressant :

[http://www.tutorialspoint.com/unix\\_commands/keytool.htm](http://www.tutorialspoint.com/unix_commands/keytool.htm)

La clé se présente sous forme de fichier avec une extension `.keystore`. La clé qu'on a générée est nommée `challenge-app-key.keystore` et le mot de passe qui lui est associé est : *passer*. Ce mot de passe vous sera demandé dans le processus de coompileation.

Ce fichier est enregistré dans le dossier contenant le projet.

### Etapes de la compilation dans un environnement Linux (ubuntu)

1. Éditer le fichier `config.xml` qui se trouve dans le dossier du projet s'il s'agit d'une première compilation. Dans ce cas, il faudra modifier certaines informations comme le nom, la version, l'id, la description... Pour mieux comprendre l'utilité de ce fichier et comment le manipuler, voici un tutoriel :  
[https://cordova.apache.org/docs/fr/latest/config\\_ref/](https://cordova.apache.org/docs/fr/latest/config_ref/)
2. Se placer dans le dossier de projet :  

```
$ cd challenge
```
3. Exécuter la commande suivante :  

```
$ cordova build --release android
```
4. Se placer dans le répertoire contenant le fichier `apk` généré :  

```
$ cd platforms/android/build/outputs/apk/
```
5. Signer le fichier `apk` avec la clé en exécutant la commande suivante  

```
$ jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore challenge-app-key.keystore android-release-unsigned.apk alias_name
```
6. Compresser le fichier `apk` avec la commande suivante (il faut que la bibliothèque `zipalign` soit installé, comme mentionné dans le tutoriel indiqué au début de cette section) :  

```
$ zipalign -v 4 android-release-unsigned.apk ChallengeECL.apk
```
7. A cette étape, la compilation est finie et le fichier `apk` peut être soumis dans le PlayStore. Par contre, il est préférable de le déplacer d'abord dans un dossier plus accessible, comme le Bureau par exemple :  

```
$ cp ChallengeECL.apk /home/Bureau/
```

### Soumission sur le PlayStore

Pour soumettre l'application sur le PlayStore, il faudra d'abord acheter un compte développeur chez Google :

<https://support.google.com/googleplay/android-developer/answer/6112435?hl=fr>

Ensuite pour le reste voici un tutoriel qui explique dans les moindres détails la procédure à suivre :

<https://electricnews.fr/publier-un-app-sous-android-petite-presentation-de-google->

## G.2.2 Compilation sous iOS

La première condition est d'avoir un Mac avec la dernière version de logiciel XCode.

La deuxième étape est d'acheter un compte développeur Apple à cette adresse :  
<https://developer.apple.com>

Ensuite, il suffit de suivre ce tutoriel : <https://medium.com/waaws-thinks/deploy-an-ionic-app-.ms8k9itgg> pour obtenir toutes les informations nécessaires.

## G.3 Description du Backend

Le Backend d'une application est sa partie immergée. Invisible pour les utilisateurs, elle représente la base de l'application avec notamment les communications avec le serveur et la base de données.

A la différence du paragraphe précédent sur le développement du frontend, nous avons ici décidé d'expliquer comment nous avons mis en place le backend. Il ne s'agit plus d'un tutoriel, comme cela a pu être le cas pour le frontend, mais plutôt d'une partie descriptive et explicative qui ne constitue donc pas un guide pour mettre en place un backend. Nous ferons néanmoins référence aux tutoriels que nous avons suivis.

### G.3.1 Description de l'API

Pour rappel, le serveur qui héberge le backend est une machine virtuelle (VM) avec un système d'exploitation *debian8*. Cette VM est hébergée sous le nom `challenge2016` dans une machine physique (Artemis) chez ÉCLAIR. L'adresse IP publique de cette VM est `challenge2016.eclair.ec-lyon.fr` (ou `156.18.24.133`).

Il est également nécessaire d'installer les logiciels de base MySQL, php et Apache2. On exécute la commande :

```
sudo apt-get install mysql-server-5.5 apache2 php5-mysql
```

Le lien suivant donne plus d'informations à ce sujet :  
[http://olange.developpez.com/articles/debian/installation-serveur-dedie/?page=page\\_2](http://olange.developpez.com/articles/debian/installation-serveur-dedie/?page=page_2)

Pour l'implémentation de la base de données, on a utilisé MySQL comme SGBD et la base a été nommée `challenge2016`.

Cette base de données a été pensée avec onze tables : Calendrier, Ecoles, Equipes, Faq, Infos, Matches, MatchesSpécial, Photo, Plans, Urgences, Users.

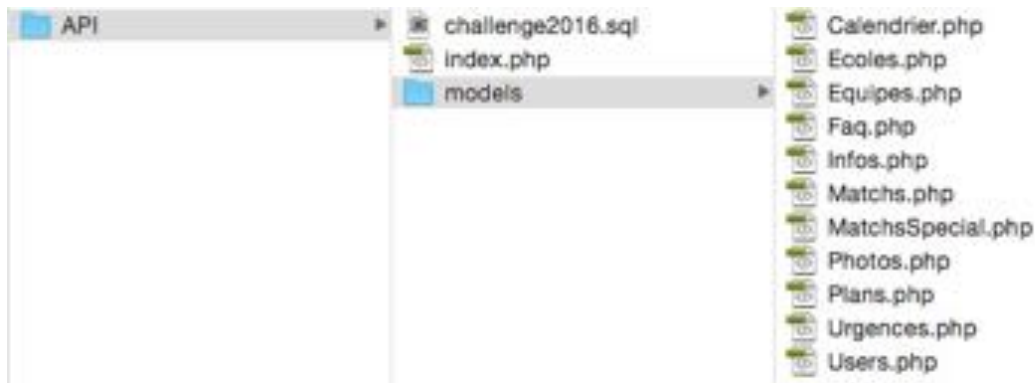


FIGURE G.10 – Fichiers de l'API

### G.3.2 Requêtes traitées par l'API

Conçu avec le framework phalcon, cet API assure le fonctionnement dynamique de l'application. Le rôle de cet API est de recevoir des requêtes, les traiter et de renvoyer les résultats issus de ce traitement.

La Figure G.11 décrit de façon exhaustive toutes les requêtes que l'API peut traiter.

### G.3.3 Le coeur de l'API : index.php

Parmi ces fichiers, le plus important est `index.php` qui contient l'ensemble du programme permettant de réaliser toutes ces requêtes.

**Pour des raisons de sécurité pour l'application, nous ne pouvons rendre compte de l'intégralité du code de ce fichier `index.php`**



Verbe	Chemin (ou endpoint)	Paramètres	Résultats renvoyés
GET	challenge-2016.eclair.ec-lyon.fr/api/users		Liste des utilisateurs
POST	challenge-2016.eclair.ec-lyon.fr/api/user	nom,login,mdp,profil	Enregistre un utilisateur
PUT	challenge-2016.eclair.ec-lyon.fr/api/user	nom,login,mdp,profil	Mettre à jour un utilisateur
DELETE	challenge-2016.eclair.ec-lyon.fr/api/user	id_user	supprime l'utilisateur
GET	challenge-2016.eclair.ec-lyon.fr/api/ecoles		Liste des écoles
GET	challenge-2016.eclair.ec-lyon.fr/api/ecoles	abrev_ecole	L'école concernée
GET	challenge-2016.eclair.ec-lyon.fr/api/ecoles/classement		Le classement des écoles
GET	challenge-2016.eclair.ec-lyon.fr/api/matches	sport,genre	Tous les matchs du sport et du genre concerné
POST	challenge-2016.eclair.ec-lyon.fr/api/matches	id_match,scores,équipes...	Modifie le match concerné
POST	challenge-2016.eclair.ec-lyon.fr/api/matches/special	id_match,scores,équipes...	Modifie le match spécial concerné
GET	challenge-2016.eclair.ec-lyon.fr/api/home/galerie		Tous les photos de la galerie
GET	challenge-2016.eclair.ec-lyon.fr/api/home/infos		Tous les news postés
GET	challenge-2016.eclair.ec-lyon.fr/api/home/galerie/liens		Les 3 dernières photos postée
GET	challenge-2016.eclair.ec-lyon.fr/api/informations/calendrier		Le calendrier du weekend challenge
GET	challenge-2016.eclair.ec-lyon.fr/api/informations/plan		Le plan des lieux sportifs
GET	challenge-2016.eclair.ec-lyon.fr/api/informations/urgence		Liste des responsables et des numéros d'urgence
GET	challenge-2016.eclair.ec-lyon.fr/api/informations/faq		Le FAQ (questions/réponses)

FIGURE G.11 – Liste des requêtes traitées par l'API

Dans le fichier `index.php`, on retrouve exactement une structure similaire à celle du tableau descriptif ci-dessus.

On va retrouver trois grandes parties :

- Une première partie (environ 30 lignes) qui permet d'importer les bibliothèques de Phalcon et de charger le dossier `models`. Elle permet d'initialiser l'API.
- Une deuxième partie (environ 90 lignes) où sont définies certaines fonctions qu'on va utiliser par la suite ou qui pourront nous servir ultérieurement si on veut enrichir l'API.
- Une troisième partie (presque 1000 lignes) dans laquelle sont définies les routes et les actions qui permettent de traiter une requête. C'est justement à ce niveau là qu'on verra la similarité entre le tableau descriptif de l'API et son implémentation.

Voici un certains nombre d'exemples de bouts de codes de cette partie qui permettent de réaliser les actions définies dans le tableau précédent.

### Exemple 1

Requête : Récupérer toutes les écoles

GET	challenge-2016.eclair.ec-lyon.fr/api/ecoles		Liste des écoles
-----	---	--	------------------

Code correspondant :

```
$app->get('/api/ecoles', function () use ($app) {
    $phql = "SELECT * FROM Ecoles ORDER BY id_ecole";
    $ecoles = $app->modelsManager->executeQuery($phql);
    $response=new Response();
    $response->setStatusCode(200, "OK");
    $data = array();
    foreach ($ecoles as $ecole) {
        $data[] = array(
            'id_ecole' => $ecole->id_ecole,
            'nom_ecole' => $ecole->nom_ecole,
            'abreviation' => $ecole->abreviation,
            'points' => $ecole->points,
            'etat_ecole' => $ecole->etat_ecole,
            'logo_ecole' => EncoderImage($ecole->logo_ecole)
        );
    }
    $response->setJsonContent(
        array(
            'status' => 'OK',
            'data' => $data
        )
    );
    return $response;
});
```

**Exemple 2** : certaines routes, sans le code complet

Requête : Ajouter un nouvel utilisateur (avec POST) Code :

POST	challenge-2016.eclair.ec-lyon.fr/api/user	nom,login,mdp ,profil	Enregistre un utilisateur
------	---	--------------------------	---------------------------

```
$app->post('/api/user', function () use ($app) {
//Code
})
```

Requête : Mettre à jour un utilisateur à partir de son id (avec PUT)

PUT	challenge-2016.eclair.ec-lyon.fr/api/user	nom,login,mdp ,profil	Mettre à jour un utilisateur
-----	---	--------------------------	------------------------------

Code :

```
$app->put('/api/users/{id:[0-9]+}', function ($id) use ($app) {
//Code
})
```

Requête : Supprimer un utilisateur à partir de son id (avec DELETE)

DELETE	challenge-2016.eclair.ec-lyon.fr/api/user	id_user	supprime l'utilisateur
--------	---	---------	------------------------

Code :

```
$app->delete('/api/users/{id:[0-9]+}', function ($id) use ($app) {
//Code
})
```

### G.3.4 Le dossier models

Le dossier `models` contient des fichiers php qui permettent d'accéder aux tables de la base de données comme si c'étaient des objets. On remarque d'ailleurs que chaque table de notre base de données dispose d'un fichier `.php` qui porte le même nom qu'elle. Le model permet d'implémenter quelques règles en utilisant des "validateurs" qui vont garantir que les données manipulées par l'API et la base de données respecteront toujours les conditions nécessaires pour l'application.

Voici comme exemple le contenu du fichier `Users.php`.

```
<?php
use Phalcon\Mvc\Model;
use Phalcon\Mvc\Model\Message;
use Phalcon\Mvc\Model\Validator\Uniqueness;

class Users extends Model
{
public function getSource(){
```

```

return 'Users';
}

public function validation()
{

    // User login must be unique
    $this->validate(
    new Uniqueness(
    array(
    "field"    => "login",
    )
    )
    );

    // Check if any messages have been produced
    if ($this->validationHasFailed() == true) {
    return false;
    }
    }
    }
}

```

Par ailleurs, un fichier caché (non visible directement) se trouve aussi dans le répertoire du projet, il s'agit du fichier texte `.htaccess` qui contient toutes les règles de réécriture d'URL pour notre fichier `index.php`. Pour le voir sous Linux par exemple, il faut se placer dans le répertoire et exécuter la commande `$ls -la`.

## G.4 Conclusion

Le manuel technique ci-dessus n'est pas exhaustif mais donne un très bon aperçu de ce à quoi il faut s'attendre pour coder l'application. Pour plus d'informations à propos du code, ne pas hésiter à contacter l'ancienne équipe ou à chercher sur internet (les forums notamment).

# Bibliographie

- [1] <http://www.dictionnaireduweb.com/framework/>
- [2] <http://www.ecommercemag.fr/Definitions-Glossaire/newsfeed-239465.html>
- [3] <http://www.dictionnaireduweb.com/api-interface-de-programmation/#more-1006>
- [4] <http://blog.nicolashachet.com/niveaux/confirmelarchitecture-rest-expliquee-en-5-regles/>
- [5] <http://sql.sh/sghd>
- [6] <http://bwp-necromance.developpez.com/tutoriel/adobe/illustrator/cs4/>
- [7] <http://ux-fr.com/experience-utilisateur-definition/>
- [8] [https://fr.wikipedia.org/wiki/Phalcon\\_\(framework\)](https://fr.wikipedia.org/wiki/Phalcon_(framework))
- [9] <https://openclassrooms.com/courses/administrez-vos-bases-de-donnees-avec-mysql>
- [10] [https://fr.wikipedia.org/wiki/Modèle\\_vue\\_contrôleur](https://fr.wikipedia.org/wiki/Modèle_vue_contrôleur) et <https://openclassrooms.com/courses/developpez-votre-site-web-avec-le-framework-le-fonctionnement-de-django>
- [11] [http://olange.developpez.com/articles/debian/installation-serveur-dedie/?page=page\\_2](http://olange.developpez.com/articles/debian/installation-serveur-dedie/?page=page_2)
- [12] <http://www.lesnumeriques.com/mobilite/parts-marche-os-mobiles-ios-ne-maintient.html>

**Check-list de rapport de Projet d'Etudes**  
**A remplir par les rédacteurs (élèves)**  
**et à insérer en dernière page du rapport**

**A développer**

Renseigner la case par le nom du responsable, ou la date ou une simple croix lorsque la vérification a été faite.

Vérification présence	Vérification qualité
-----------------------	----------------------

**Contenu**

Résumé en français	OK	OK
Résumé en anglais	OK	OK
Table des matières	OK	OK
Table des figures	OK	OK
Introduction	OK	OK
Conclusion générale	OK	OK
Bibliographie	OK	OK
Citation des références dans le texte	OK	OK

**Forme**

Vérification orthographe		OK
Pagination		OK
Homogénéité de la mise en page		OK
Lisibilité des figures		OK