



Site et appli FootRegion

Documentation technique

Sommaire

Cahier des Charges	2
Organisation du projet FootRegion	4
Spécifications fonctionnelles	10
Spécifications techniques	21
Conception de la solution	23
Codage du composant com_footregion	27
Application AndroidFootRegion	30
Intégration et tests d'intégration du site FootRegion	33
Tests et recette fonctionnelle	35
Déploiement	36

Cahier des Charges

Présentation de la FFF

La Fédération française de football (FFF)...

La FFF compte aujourd'hui près de 2,2 millions de licenciés dont 400 000 bénévoles et 160 000 féminines.

- 2 160 788 licences dont 159 128 féminines (au 30 juin 2017).
- 14 993 clubs dont 40 clubs professionnels participant aux championnats de L1 (20 clubs) et de L2 (20 clubs).
- 836 135 matches par an.
- 13 ligues régionales + 9 d'outre-mer.
- 90 districts.
- 7 000 salariés.
- 400 000 bénévoles.

(A compléter par Saïd)

Objectif du projet FootRegion

L'objectif du projet FootRegion est de développer un site Web et une application mobile...

(A compléter par Allan)

Analyse de l'existant

La FFF possède son propre site...

(A compléter par Quentin)

Analyse des besoins

Les besoins de la solution FootRegion sont les suivants :

- Un site Web donc la page d'accueil et les pages de présentation de la FFF seront publiques. L'accès aux autres pages (discussions, matchs, joueurs, etc.) sera réservé aux utilisateurs enregistrés.
- Une application mobile, sur Android dans un premier temps, pour permettre aux utilisateurs d'accéder à quelques services en ligne.

- Les utilisateurs du site FootRegion seront répartis en quatre catégories : directeur, entraîneur, joueur, arbitre.
- L'administrateur du site FootRegion gèrera les demandes d'inscription des utilisateurs et leurs rôles déterminant leurs droits d'accès au site :
 - Chaque membre peut gérer son profil, lancer et participer à une discussion, poster des messages, consulter les matchs et les tournois.
 - Les directeurs de clubs peuvent...
 - Les entraîneurs peuvent modifier...
 - Les joueurs peuvent...
 - Les arbitres peuvent...
- D'autres services pourront être proposés comme :
 - ...

(A compléter par Rubean)

Organisation du projet FootRegion

Organisation générale du projet

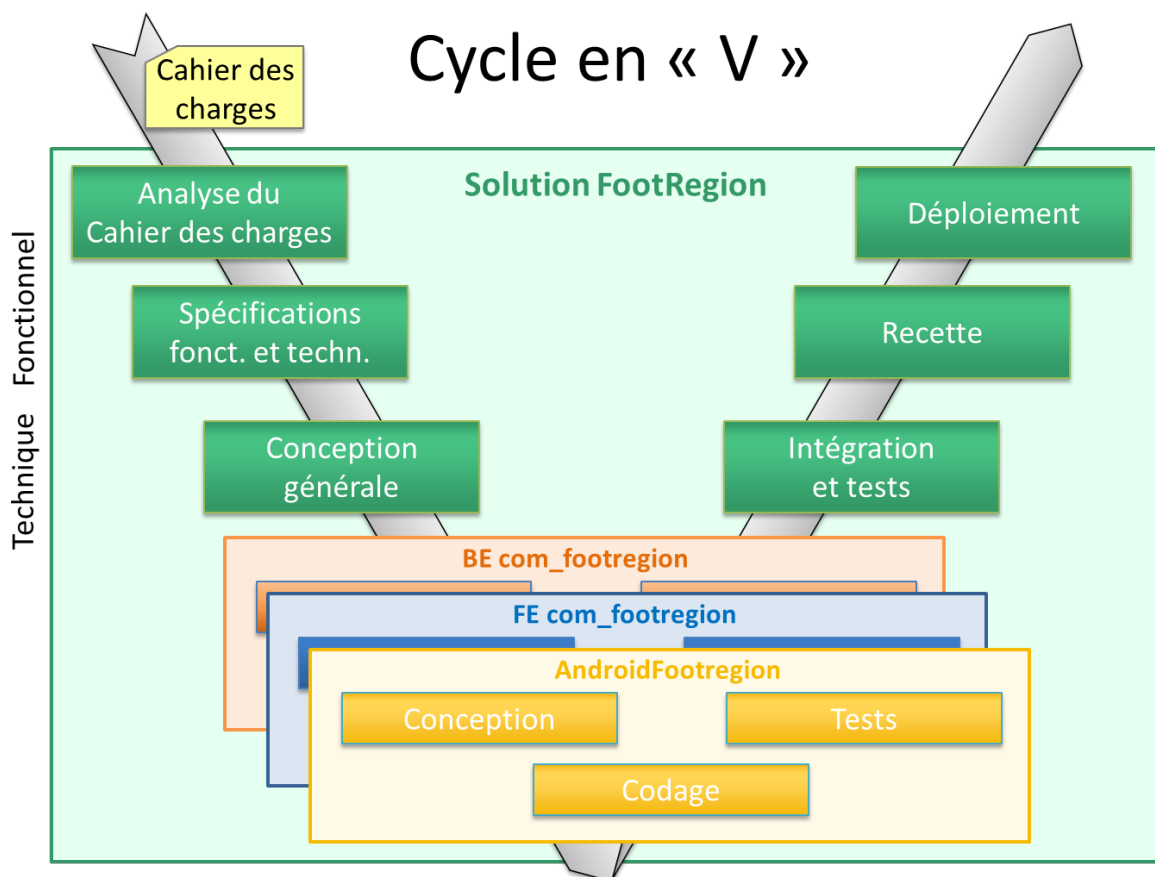
L'organisation du projet FootRegion mise en place par le professeur en charge des modules SLAM4, SLAM5 et PPE3-4, a coïncidé avec la progression pédagogique établie pour ces modules, le projet FootRegion servant de fil conducteur au déroulé des cours, travaux pratiques et PPE.

Le projet FootRegion a d'abord été décomposé en trois livrables principaux liés à l'architecture du framework Joomla retenu et présenté plus loin (cf. chapitre conception et architecture) :

- Le **site d'administration**, avec le développement de la partie backend du composant com_footregion (extension Joomla spécialement développée pour le projet).
- Le **site public** avec accès réservé aux adhérents (directeur, entraîneur, joueur, etc.) avec la configuration du CMS Joomla et le développement de la partie frontend du composant com_footregion.
- Le développement de l'**application mobile Android** et de l'API associée.

Approche méthodologique

Le développement s'est fait selon la méthodologie du cycle de vie des logiciels (cycle en « V ») définissant les grandes étapes du projet depuis l'analyse du cahier des charges jusqu'à la recette fonctionnelle et le déploiement :



Le cycle en « V » de la solution FootRegion commence à partir du cahier des charges, avec les trois étapes suivantes :

- Analyse du cahier des charges.
- Spécifications fonctionnelles et techniques.
- Conception générale.

A partir de cette étape, l'architecture générale de la solution est définie et trois développements sont identifiés :

1. Le backend du composant com_footregion.
2. Le frontend du composant com_footregion.
3. L'application mobile AndroidFootRegion.

Ces trois développements se décomposent en trois sous-cycles en « V » :

- Conception
- Codage
- Test

Une fois ces trois développements terminés, le cycle en « V » principal continue, avec les trois étapes suivantes :

- Intégration et tests.
- Recette.
- Déploiement.

Le déploiement aboutit dans ce projet école à la mise en production d'une solution Web et mobile prototype, sur des serveurs accessibles par Internet.

Planning général

Le planning général du projet FootRegion et de ses trois sous-projets s'est inscrit dans le calendrier (progression pédagogique) mis en place sur l'année scolaire 2018-2019 comme suit :

Semaine de cours			Etape	Cours + TP (12 h)
N°	Lu	Sa		
0	3/9	8/9	Analyser la demande	1 - Cahier des charges
1	10/9	15/9		
2	17/9	22/9	Choisir une solution	2 - Spécifications technique et fonctionnelle
3	24/9	29/9		
4	1/10	6/10	Concevoir la solution	3 - Architecture de la solution applicative
5	8/10	13/10		
6	15/10	20/10	Concevoir le BE	4a - Architecture du BE du composant
Vac. Toussaint				
7	5/11	10/11	Coder le BE	5a - Codage du BE du composant
8	12/11	17/11	Concevoir le FE	4b - Architecture du FE du composant
9	19/11	24/11	Coder le FE	5b - Codage du FE du composant
10	26/11	1/12		
11	3/12	8/12		
12	10/12	15/12		
13	17/12	22/12		
Vac. Noël				
14	7/1	12/1		Stage en entreprise (6-8 sem.)
15	14/1	19/1		
16	21/1	26/1		
17	28/1	2/2		
18	4/2	9/2		
19	11/2	16/2	Concevoir l'AM	
20	18/2	23/2	Coder l'AM	4c - Architecture de l'application mobile
				5c - Codage de l'application mobile
Vac. Hiver				
21	11/3	16/3	Coder le FE	5b - Codage du FE du composant
22	18/3	23/3		
23	25/3	30/3	BTS blanc	
24	1/4	6/4	Coder l'AM	5c - Codage de l'application mobile
25	8/4	13/4	Intégrer et tester	7 - Intégration et tests de la solution
26	15/4	20/4	Documenter	8 - Documentation technique de la solution

Gestion du projet

Des groupes de travail ont été constitués pour faciliter le partage de connaissance et la collaboration entre les étudiants mis en situation professionnelle lors des travaux pratiques.

Des contributions individuelles au projet ont été aussi demandées à chaque étudiant pour permettre à chacun de développer son autonomie.

Les tableaux suivants identifient, pour le backend puis pour le frontend, les **responsables par domaine et table** (ou classes métier du modèle des données, telles que définies plus bas dans le chapitre des spécifications fonctionnelles) :

Backend de com_footregion :

Domaine	Table	MVC Liste	MVC Détail
Utilisateur	utilisateurs	Saïd	Serinkan
	discussions	Saïd	Serinkan
	messages	Serinkan	Saïd
Club	clubs	Sébastien	Alexandre
	directeurs	Sébastien	Alexandre
	categories	Sébastien	Alexandre
Equipe	equipes	Rubean	Morgan
	entraîneurs	Rubean	Morgan
	joueurs	Rubean	Morgan
Match	matches	Quentin	Quentin
	tournois	Romain	Romain
	statuts	Quentin	Romain
Arbitre	arbitres	Allan	Armand
	matches_arbitres	Allan	Armand
	signalements	Allan	Armand

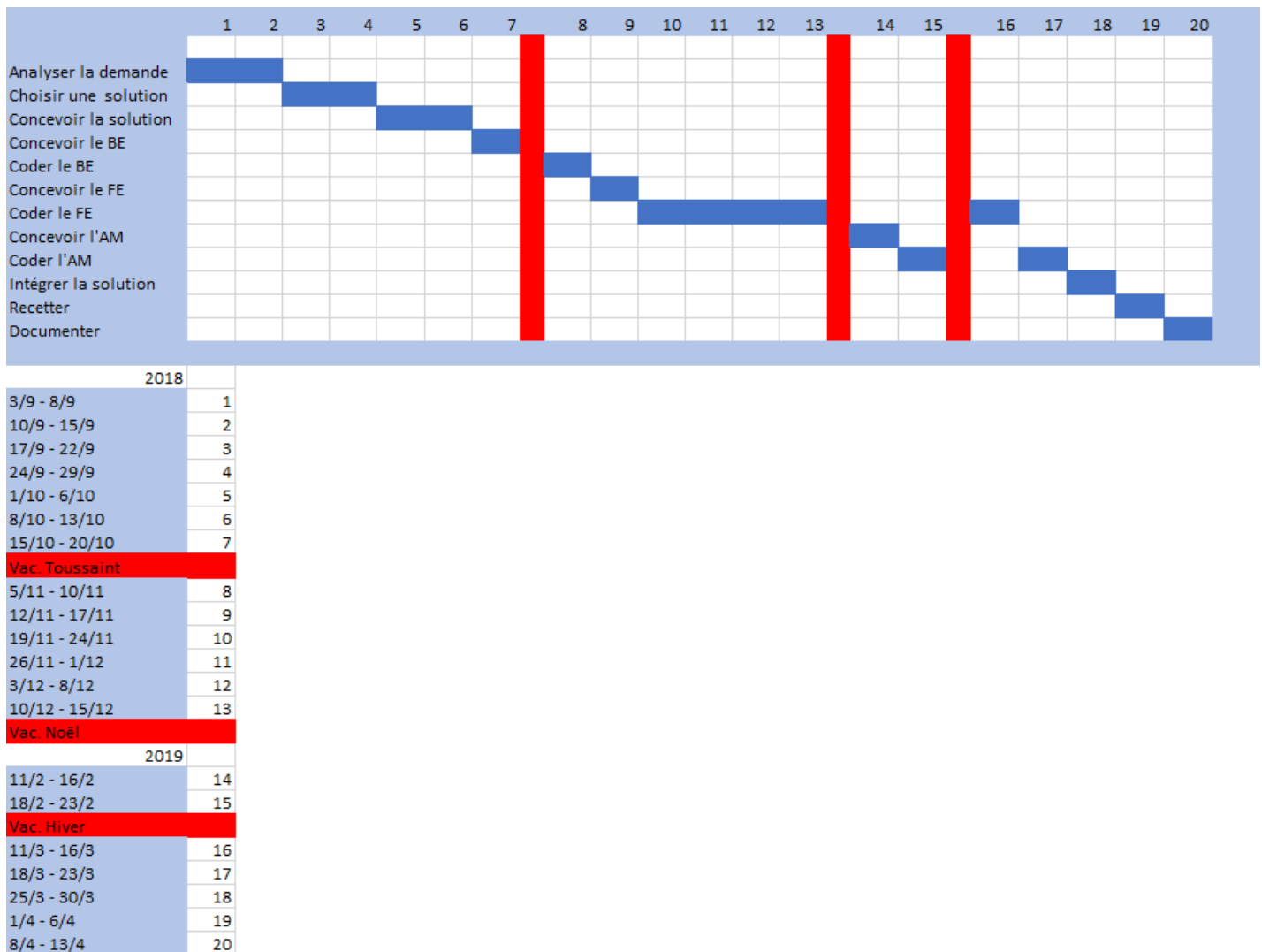
Frontend de com_footregion :

Domaine	Rubriques	Directeur	Entraîneur	Joueur	Arbitre
Utilisateur	Mon profil	Serinkan			
	Discussions	Saïd			
Match	Matches	Quentin			
	Tournois	Romain			
Club	Mon club	Sébastien			
	Clubs	Alexandre			
Equipe	Equipes	Rubean			
	Joueurs	Morgan			
	Entraîneurs	Morgan			
Arbitre	Arbitrages				Allan
	Signalements				Armand
Intégration		<i>Sébastien</i> <i>Alexandre</i>	<i>Morgan</i> <i>Allan</i>	<i>Rubean</i> <i>Armand</i>	<i>Quentin</i> <i>Romain</i>

AndroidFootRegion :

Acteurs	Rubriques	Vue liste	Vue détail
Tous	Discussions	Saïd	Serinkan
	Matches	Romain	Quentin
Directeur	Mes clubs	Sébastien	Alexandre
Entraîneur	Mes équipes	Rubean	Morgan
Joueur	Mon équipe	Allan	Armand

Diagramme de Gantt :



Gestion des versions logicielles

La gestion des versions logicielles est faite avec le logiciel libre **Git** qui permet :

- D'enregistrer des **états de la base de code** appelés commits (des versions) associés à des commentaires, permettant de suivre plus facilement l'évolution du projet et de revenir en arrière si besoin.
- D'organiser les versions dans des **branches**, permettant de développer une version sans impacter les autres.
- De **travailler à plusieurs** sur le même projet, grâce à un système de merge (fusion).
- De disposer de **sauvegardes** du projet grâce à une logique de distribution. Car chaque ordinateur dispose du projet dans son intégralité.

Le système d'identification des utilisateurs de Git fonctionne avec une simple adresse email.

Chaque utilisateur peut créer des commits à chaque fonctionnalité développée.

Pour simplifier le travail à plusieurs, un dépôt partagé en ligne est également créé sur le site **github.com**, d'accès libre pour les dépôts publics.

Chaque utilisateur peut ainsi envoyer ses commits sur github et fusionner son code avec celui des autres.

Le système de branche n'a pas été développé, il n'y en a donc qu'une seule : la branche **master**.

Spécifications fonctionnelles

Le site FootRegion se décompose en deux parties :

- Le **backend FootRegion**, site d'administration du site FootRegion.
- Le **frontend FootRegion**, site FootRegion constitué d'une partie publique et d'une partie à accès réservé aux utilisateurs enregistrés.

Les présentes spécifications fonctionnelles définissent les acteurs, les diagrammes de cas d'utilisation et les diagrammes de séquence associés, les interfaces utilisateurs et le modèle des données commun aux deux sites.

Acteurs

Au sens UML, les acteurs concernés par le système FootRegion sont les suivants :

- Backend FootRegion: administrateur.
- Frontend FootRegion: directeur, entraîneur, joueur et arbitre.

Diagramme de cas d'utilisation du backend FootRegion

Le diagramme des cas d'utilisation UML du backend FootRegion est le suivant :

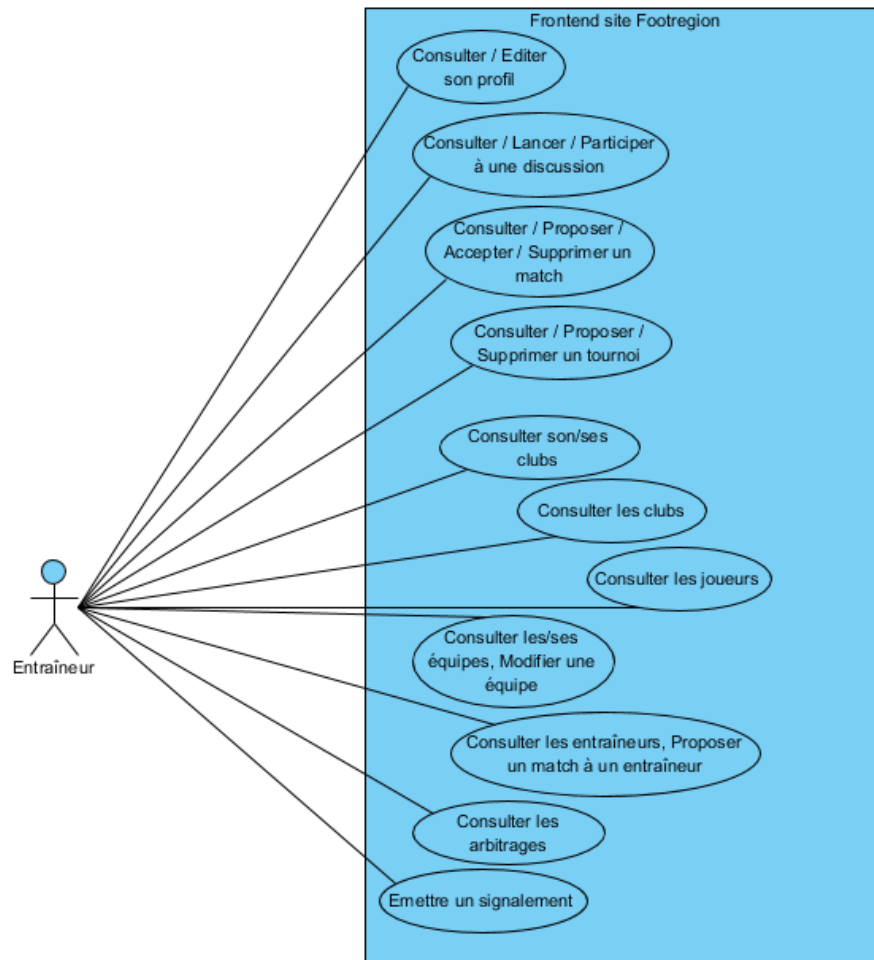


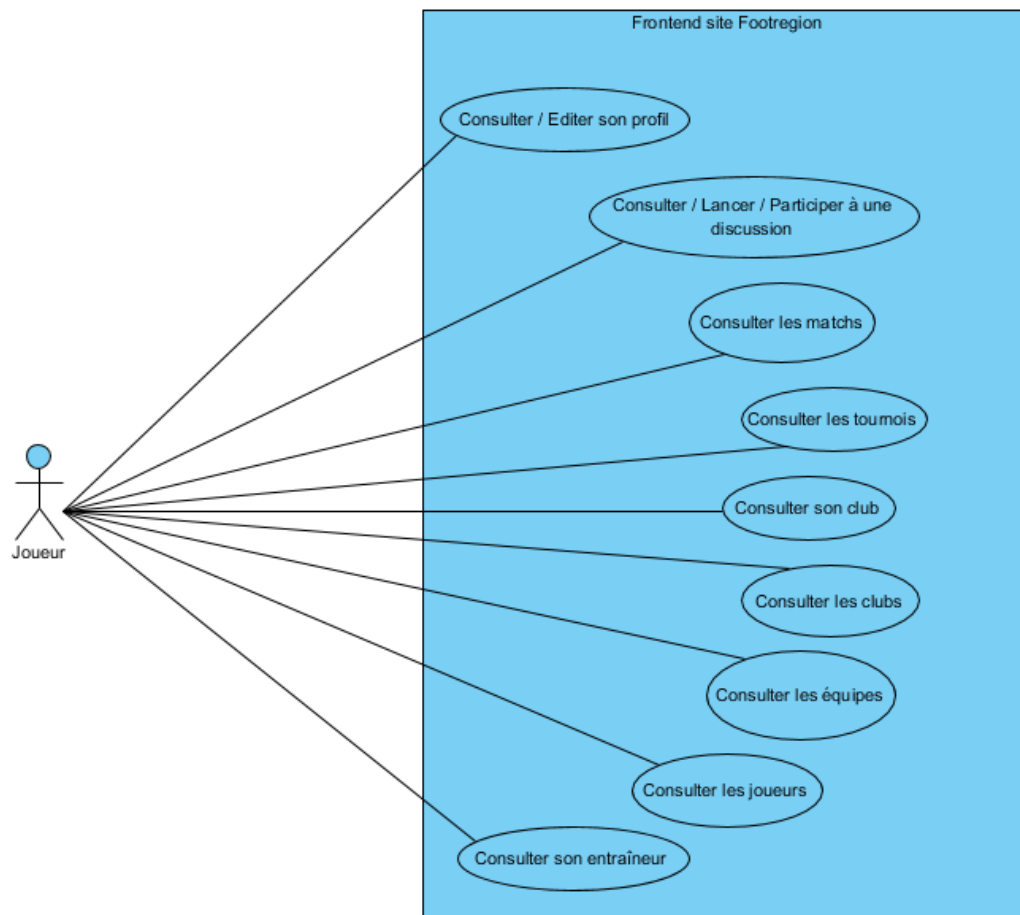
« Gérer... » inclut les 5 cas d'utilisation : Ajouter, Lire, Modifier, Supprimer et Publier/Dépublier.

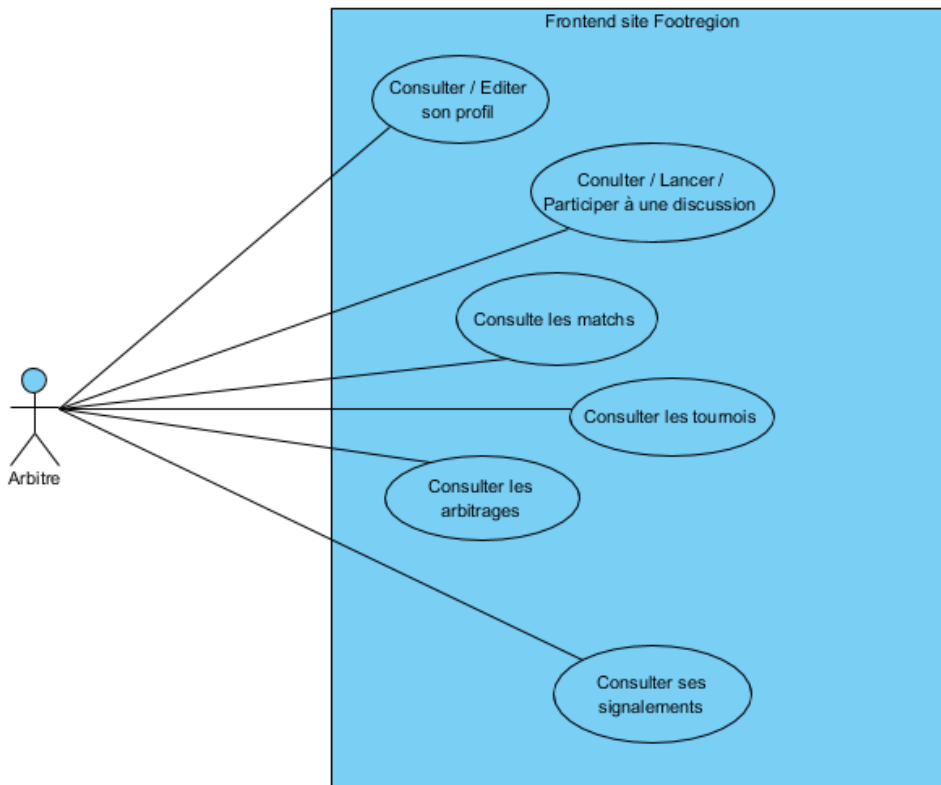
Diagramme de cas d'utilisation du frontend FootRegion

Les diagrammes des cas d'utilisation du frontend FootRegion sont les suivants :





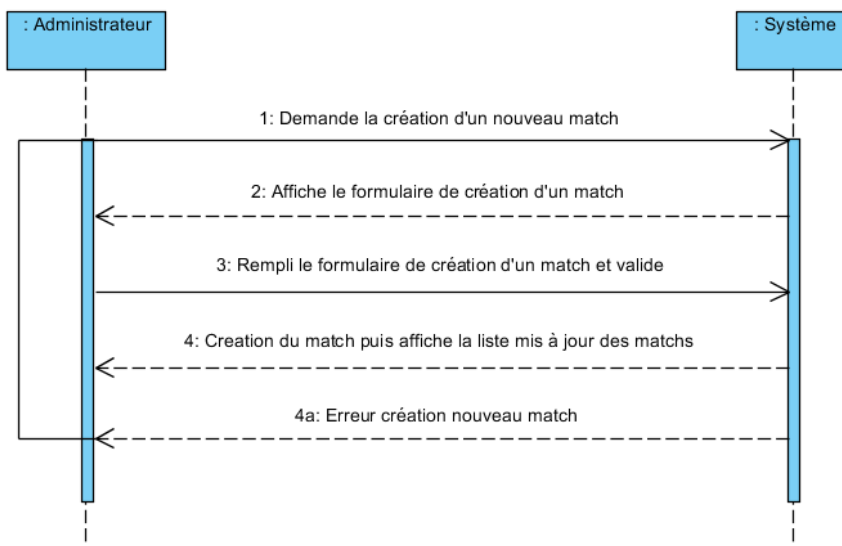


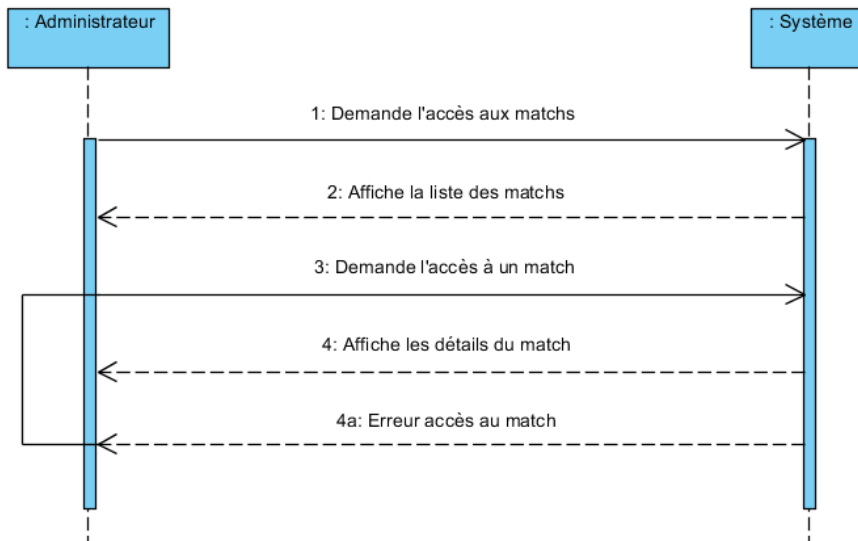


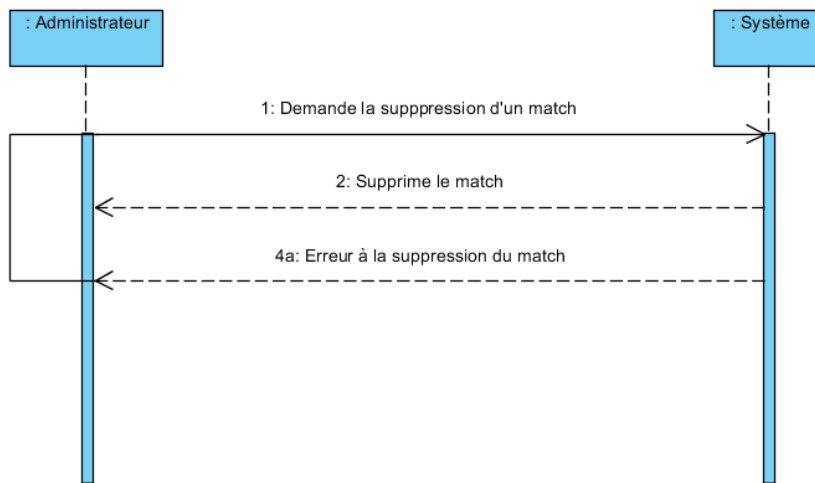
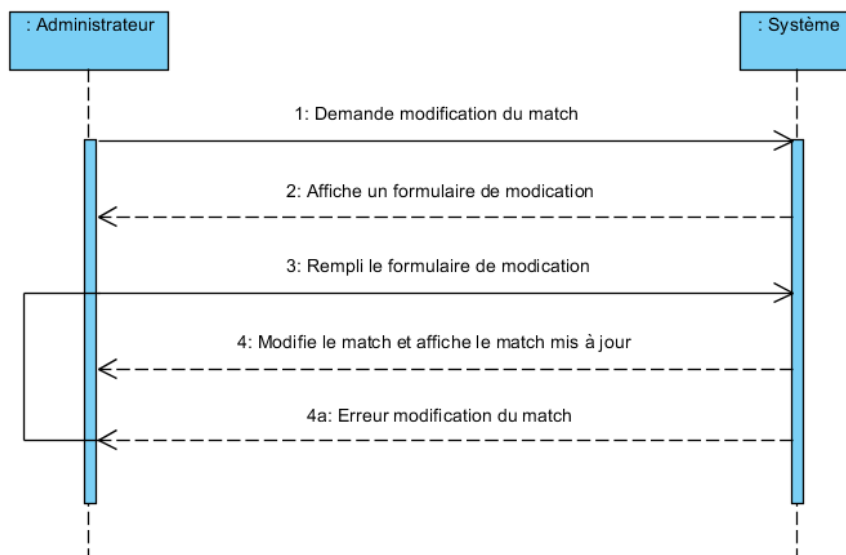
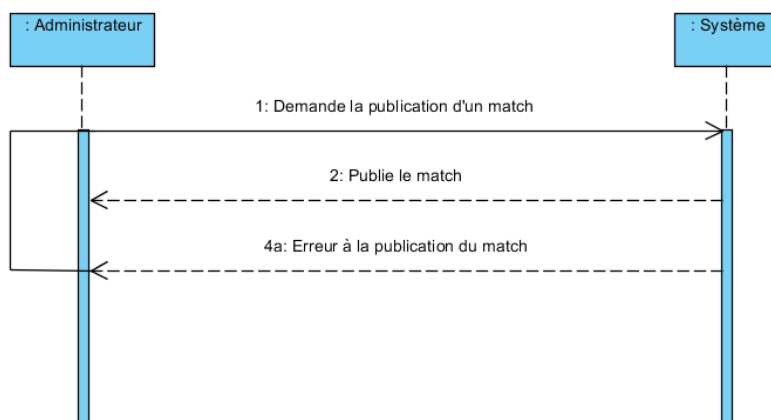
Diagrammes de séquence du backend FootRegion

Les diagrammes de séquence du backend FootRegion associés aux cas d'utilisation du backend définis plus haut ont été ensuite développés en notation UML 2. En voici quelques exemples :

Ajouter un match :



Consulter un match :

Supprimer un match :**Modifier un match :****Publier/Dépublier un match :**

Diagrammes de séquence du frontend FootRegion

Les diagrammes de séquence du frontend FootRegion associés aux cas d'utilisation du frontend définis plus haut ont été ensuite développés en notation UML 2. En voici quelques exemples :

(A compléter par Morgan)

Interfaces utilisateurs du backend FootRegion

Voici la page d'accueil du backend FootRegion :

(A compléter par Saïd)

Interfaces utilisateurs du frontend FootRegion

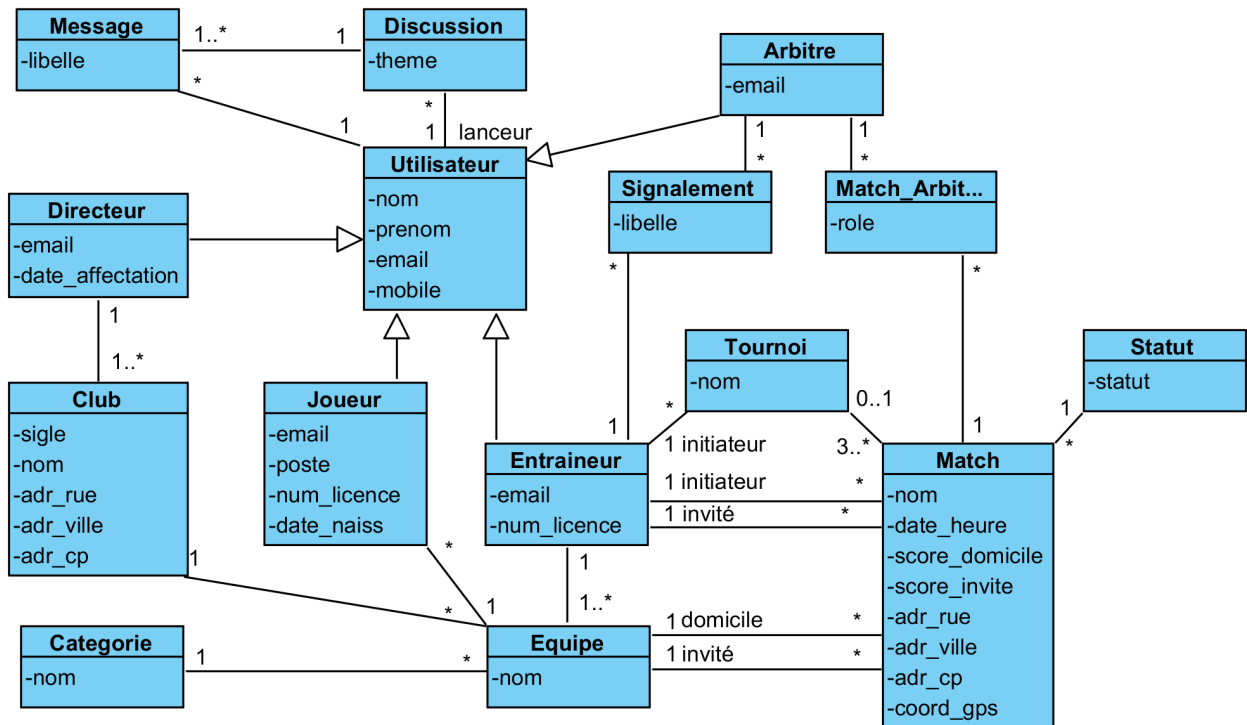
La page d'accueil du frontend FootRegion se présente comme suit :

Après connexion, l'espace directeur apparaît comme suit :

(A compléter par Saïd)

Modèle des données FootRegion

Le modèle des données FootRegion est décrit par le diagramme de classes UML suivant :



L'implémentation de ce modèle dans le SGBD MySQL avec l'outil concepteur de PhpMyAdmin donne le diagramme suivant (structure de la base de données FootRegion) :

(A compléter par Sébastien)

Spécifications techniques

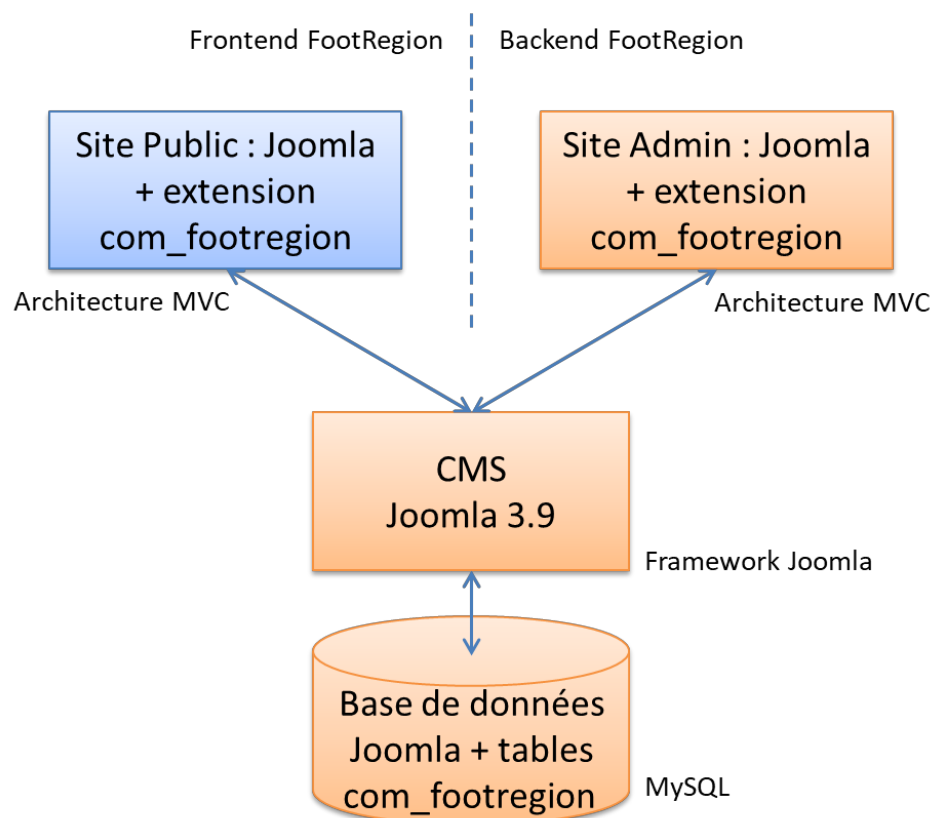
Framework et CMS Joomla

Afin de réduire les risques liés à la maîtrise d'œuvre d'un tel projet, l'équipe projet FootRegion a retenu le principe d'une architecture logicielle ouverte (architecture MVC : Modèle-Vue-Contrôleur) avec Framework/CMS Joomla 3.6 et un développement PHP en deux phases :

1. Développement du backend FootRegion et de la base de données « FootRegion » sous la forme d'un composant « com_footregion » installable dans le CMS Joomla 3.5.
2. Développement du frontend FootRegion avec le composant « com_footregion » pour implémenter les fonctionnalités attendues des utilisateurs.

La technologie Joomla a été choisie parmi les trois CMS (Content Management System : outil de gestion de contenu sur Internet) open-source et gratuits les plus populaires du marché : WordPress, Drupal et Joomla. Sa simplicité d'utilisation, la qualité de ses extensions et le dynamisme de ses communautés d'utilisateurs et de développeurs en France, en Europe et dans le monde assurent à Joomla un suivi et une évolution de très grande qualité.

Le schéma d'architecture logicielle est le suivant :



Les éléments coloriés en orange correspondent aux développements relatifs à la première phase et comprennent les deux éléments du socle commun, la base de données Joomla sous MySQL et l'installation de l'API Joomla 3.9, permettant ensuite l'intégration du composant « com_footregion » spécialement développé pour les besoins du site FootRegion. L'élément colorié en bleu correspond à la deuxième phase du développement explicitée ci-dessus.

Environnements de développement, tests et production

L'environnement de développement et de test qui a été mis en place est le suivant :

- Installation sur chaque PC étudiant de :
 - **XAMPP 7.1.9** (PHP 7.1.9, Apache 2.4.27, MariaDB 10.1.26, phpMyAdmin 4.7.4)
 - **Visual Studio Code 1.33.0**
 - **Joomla 3.9**
- Installation sur le serveur du labo SLAM :
 - Machine virtuelle Debian 8.2
 - Services Apache2, PHP et MySQL
 - Logiciel phpMyAdmin

L'environnement de production choisi a été une plateforme mutualisée de l'hébergeur OVH.com (offre Pro sur serveur LAMP) dont les détails sont donnés dans le chapitre déploiement.

(A compléter par Alexandre)

Conception de la solution

Architecture logicielle MVC

Le choix d'une architecture MVC répond aux besoins d'ouverture, de maintenance et d'évolutivité du site par une organisation standardisée du code source. Le but d'une telle architecture est de structurer le code, pour chaque cas d'utilisation ou bloc fonctionnel, en trois parties :

- Le **modèle** gère les données du site. Il récupère les informations dans la base de données, les organise et les assemble pour qu'elles puissent ensuite être traitées par le contrôleur et la vue. Cette partie contient donc les requêtes SQL organisées sous forme de fonctions PHP.
- La **vue** gère l'affichage. Elle organise la façon dont les données sont affichées à l'écran. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples.
- Le **contrôleur** gère la logique du code et les demandes utilisateurs. Le contrôleur récupère les données du modèle, les analyse/les traite, et renvoie les données à afficher à la vue. Le contrôleur contient exclusivement du code PHP, organisé sous forme de fonctions.

L'architecture du composant com_footregion respecte l'arborescence et les règles de nommage définies pour la plateforme Joomla 3.9.

Architecture du backend de com_footregion

L'arborescence des dossiers du backend du composant com_footregion est la suivante :
(dossier **footregion/administrator/components/com_footregion**)

(A compléter par Sébastien)

Les règles de nommage des fichiers php des contrôleurs, modèles et vues sont les suivantes :

- nom au pluriel pour les listes (ex : messages)
- nom au singulier pour les formulaires de détail (ex : message)

Ainsi, le dossier **controllers** comprend les fichiers php suivants :

- message.php
- messages.php
- ...

Il en va de même pour les **models**, à ceci près qu'un dossier supplémentaire est présent :
Le dossier **forms**, contenant les définitions xml des formulaires d'édition :

- message.xml
- ...

Les **vues** quant à elles ont chacune un dossier à leur nom, ex pour messages :

Le dossier **helpers** contient les éléments communs à toutes les pages (ex : le contenu du menu de gauche)

Le dossier **languages** contient les constantes de traductions pour chacune des langues supportées

Architecture du frontend de com_footregion

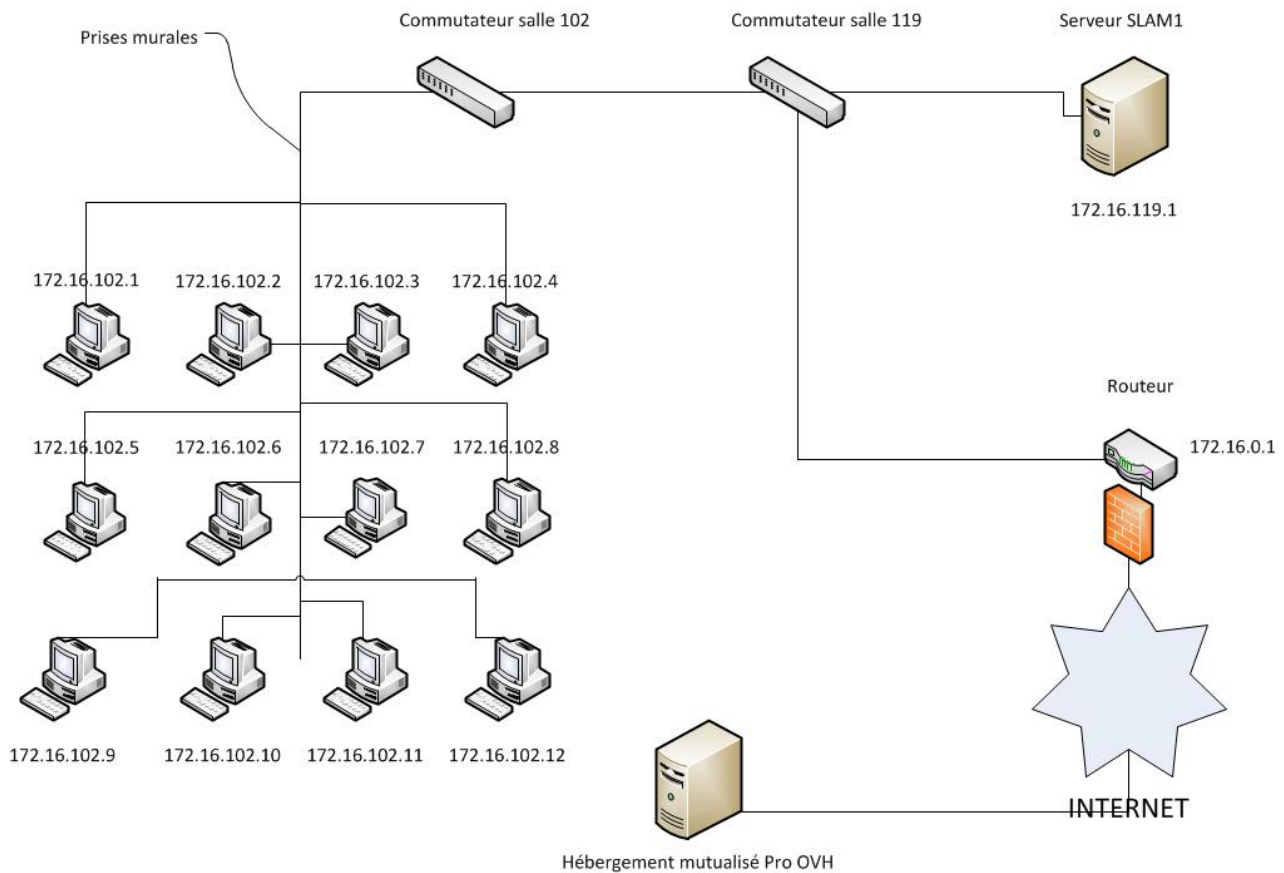
L'arborescence du frontend du composant com_footregion est la suivante :

(A compléter par Sébastien)

Le dossier **controllers** contient les fichiers suivants :

Architecture matérielle

L'infrastructure réseau mise en œuvre pour le développement de FootRegion est la suivante :



(A compléter par Alexandre)

Codage du composant com_footregion

Codage MVC de com_footregion

Les contrôleurs, modèles, tables et vues constituant le code PHP du composant com_footregion selon l'arborescence et les règles de nommage des fichiers définis plus-haut sont conçus comme des classes PHP héritant des classes correspondantes du framework Joomla : JController, JModel et JView.

Un exemple de code PHP commenté pour ces trois types de classes est présenté ci-après (liste des utilisateurs dans le backend).

(A compléter par Romain)

Contrôleur « utilisateurs.php » :

```
<?php
defined('_JEXEC') or die('Restricted access');

class FootregionControllerUtilisateurs extends JControllerAdmin
{
    // surcharge pour gérer la suppression de utilisateurs par le modèle adéquat
    public function getModel($name = 'Utilisateur', $prefix = 'ArvieModel')
    {
        // récupérer le modèle de détail ($name au sigulier) pour la suppression
        // assistée d'un (des) enregistrement(s)
        $model = parent::getModel($name, $prefix, array('ignore_request' =>
true));
        return $model;
    }
}
```

Modèle « utilisateurs.php » :

```

<?php
defined('_JEXEC') or die('Restricted access');

class FootregionModelUtilisateurs extends JModelList
{
    public function __construct($config = array())
    {
        // précise les colonnes activant le tri
        if (empty($config['filter_fields']))
        {
            $config['filter_fields'] = array(
                'id',          'u.id',
                'nom',         'u.nom',
                'prenom',      'u.prenom',
                'email',       'u.email',
                'mobile',      'u.mobile',
                'password',    'ju.password',
                'published',   'u.published',
                'created',     'u.created',
                'created_by',  'u.created_by',
                'modified',   'u.modified',
                'modified_by', 'u.modified_by',
                'hits',       'u.hits'
            );
        }
        parent::__construct($config);
    }

    protected function populateState($ordering = null, $direction = null)
    {
        // récupère les informations de la session utilisateur nécessaires au
        paramétrage de l'écran
        $search = $this->getUserStateFromRequest($this->context.'.filter.search', 'filter_search');
        $this->setState('filter.search', $search);

        $published = $this->getUserStateFromRequest($this->context.'.filter.published', 'filter_published', '');
        $this->setState('filter.published', $published);

        parent::populateState('nom', 'asc');
    }
    ...
}

```

Vue « utilisateurs/view.html.php » :

```

<?php
defined('_JEXEC') or die('Restricted access');

class FootregionViewUtilisateurs extends JViewLegacy
{
    function display($tpl = null)
    {
        // récupère la liste des items à afficher
        $this->items = $this->get('Items');
        // récupère l'objet jPagination correspondant à la liste
        $this->pagination = $this->get('Pagination');

        // récupère l'état des information de tri des colonnes
        $this->state = $this->get('State');
        $this->listOrder = $this->escape($this->state->get('list.ordering'));
        $this->listDirn = $this->escape($this->state->
>get('list.direction'));

        // récupère les paramètres du fichier de configuration config.xml
        $params = JComponentHelper::getParams('com_arvie');
        $this->paramDescShow = $params->get('jarvie_show_desc', 0);
        $this->paramDescSize = $params->get('jarvie_size_desc', 70);
        $this->paramDateFmt = $params->get('jarvie_date_fmt', "d F Y");

        // affiche les erreurs éventuellement retournées
        if (count($errors = $this->get('Errors')))
        {
            JError::raiseError(500, implode('<br />', $errors));
            return false;
        }

        // ajoute la toolbar contenant les boutons d'actions
        $this->addToolBar();
        // invoque la méthode addSubmenu du fichier de soutien (helper)
        ArvieHelper::addSubmenu('utilisateurs');
        // prépare et affiche la sidebar à gauche de la liste
        $this->prepareSideBar();
        $this->sidebar = JHtmlSidebar::render();

        // affiche les calques par appel de la méthode display() de la classe
        parent
        parent::display($tpl);
    }
    ...
}

```

Application AndroidFootRegion

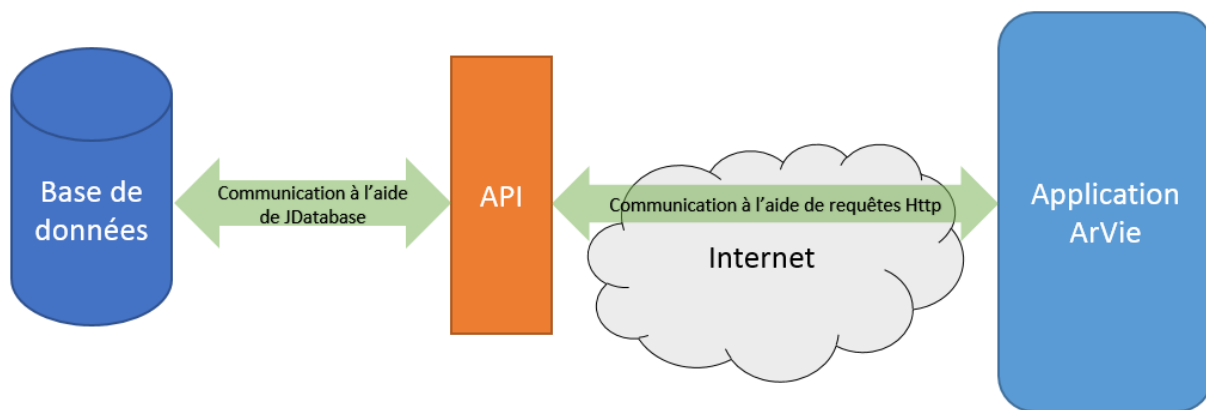
Architecture de la solution

Côté serveur, une API de test a été spécialement développée en php pour pouvoir consulter la base de données FootRegion à travers le framework Joomla par le protocole http.

Côté client, application Android a été développée en Java pour permettre à l'utilisateur « client » disposant d'une tablette ou d'un téléphone Android de se connecter à l'API par le protocole http et consulter ensuite les offres et ses candidatures.

En environnement de développement, serveur et terminal Android sont connecté au même réseau local.

(A compléter par Sébastien)



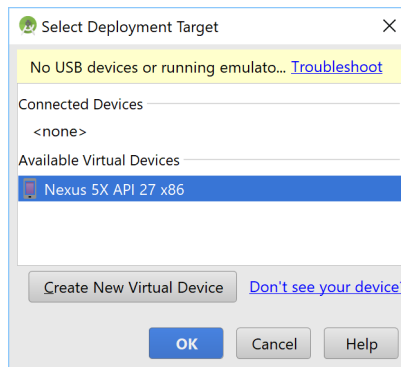
Android Studio

Conformément aux dernières recommandations de Google, l'environnement de développement Android Studio a été choisi pour concevoir et tester l'application AndroidArVie. De plus l'OS Android représente 65% des parts de marché en France ce qui n'est pas négligeable.

Android Studio est un IDE qui permet de développer des applications Android en Java, il bénéficie aussi de la technologie WYSIWYG (What You See Is What You Get) qui permet de voir et gérer la mise en page à côté de l'XML.

En phase de développement, l'application peut être exécuté localement sur un émulateur de terminal intégré à Android Studio, comme par exemple « Nexus 5X API 27 x86 » qui émule un smartphone Nexus 5X :

(A compléter par Allan)



Cas d'utilisation

Les cas d'utilisation retenus pour l'application AndroidFootRegion sont les suivants :

- 1 Se connecter.
- 2 Consulter mes messages.
- 3 Consulter joueurs.
- 4 ...

Structure et code de l'application

La structure de l'application AndroidFootRegion est la suivante :

- Le **manifest** qui permet de déclarer les activités et de définir laquelle sera lancée au lancement de l'application dans notre cas il permet de lancer la page de connexion. C'est un fichier XML dont le code est le suivant :

- Des **activités** développées en Java qui fonctionnent comme les parties contrôleurs et modèles d'une architecture MVC :

- JSONParser : permet de définir les fonctions de requête http avec l'API.
- MainActivity : affiche la vue de connexion :

Extrait du code de l'activité MainActivity qui appelle la vue « activity_main »

- mainScreenActivity : affiche la vue de menu après connexion
- MesAbonnementsActivity : Affiche la liste des abonnements de l'utilisateur
- ...

- Les **layouts** développées en XML qui permettent de définir ce qui est affiché sur le terminal :

- activity_main : Page de connexion
- activity_main_screen : Menu (mes abonnements, mes groupes)
- ...

(A compléter par Saïd) Expliquer le code avec les layouts et les ressources, faire la comparaison avec le MVC

Intégration et tests d'intégration du site FootRegion

Création du composant packagé com_footregion.zip

Il s'agit de créer un composant packagé **com_footregion.zip** installable depuis le CMS Joomla 3.9 et contenant les éléments suivants :

- Admin
 - Controllers
 - Helpers
 - Language
 - Models
 - Sql
 - Tables
 - Views
 - Acces.xml
 - Footregion.php
 - Footregion.xml
 - Config.xml
 - Controller.php
 - Index.html
- Media
 - Css
 - Images
 - Js
 - Index.html
- Footregion.xml

(A compléter par Morgan)

Test d'installation du composant packagé com_footregion.zip

Il s'agit de vérifier la bonne installation du composant **com_footregion.zip** à partir du menu du backend de Joomla : Extensions / Gérer / Installation. En cliquant sur le bouton « Choisir un fichier », on sélectionne le fichier com_arvie.zip puis on valide. La bonne installation est confirmée par un message sur fond vert : « Installation du composant réussie ».

(A compléter par Morgan)

Test de l'API

Il s'agit de tester la bonne intégration de l'API du site FootRegion et des vues correspondantes dans la base de données depuis un navigateur Internet par l'envoi de requêtes HTTP avec les url définies dans l'API et retournant le code JSON attendu.

L'utilisation du navigateur Chrome avec l'extension « JSON Viewer » permet de visualiser en 2D et en syntaxe colorée la chaine JSON retournée par l'API, comme par exemple :

(A compléter par Saïd)

Tests et recette fonctionnelle

Tests du backend FootRegion

Le cahier de tests du backend FootRegion comprend pour chaque cas d'utilisation identifié lors des spécifications une fiche de test selon modèle suivant :

(A compléter par Rubean)

Tests du frontend FootRegion

Les tests du frontend FootRegion se sont déroulés selon le même principe que ceux du backend, à partir d'un cahier de tests composé de fiches de tests associées aux cas d'utilisation spécifiés dans le frontend.

(A compléter par Rubean)

Tests de l'application AndroidFootRegion

Les tests de l'application AndroidFootRegion ont consisté à vérifier le bon affichage d'un certain nombre d'écrans comme par exemple les abonnements de M. DUPOND :

(A compléter par Romain)

Déploiement

Le déploiement du site FootRegion s'est fait sur une plateforme mutualisée LAMP hébergée chez OVH, après achat du nom de domaine disponible lla-sio.fr et de l'hébergement Pro pour un an (5,99 €HT/mois), de la façon suivante :

1. Préparation des fichiers à télécharger :
 - Export de la base de données ArVie complète au format SQL (CMS Joomla + extension com_footregion).
 - Copie du fichier footregion.sql à la racine de l'arborescence des fichiers du site egs.com.
2. Préparation de la plateforme d'hébergement :
 - Création du sous domaine footregion dans www.la-sio.fr.
 - Création du répertoire footregion à la racine www du site (correspondant au sous-domaine).
 - Création de la base de données sous MySQL.
3. Téléchargement des fichiers :
 - Téléchargement de l'arborescence des fichiers par FTP dans le répertoire www du serveur OVH.
4. Installation de la base de données :
 - Import de la base de données avec phpMyAdmin (serveur OVH).
5. Tests :
 - Test du site public à l'url : www.la-sio.fr/FootRegion
 - Test du site public à l'url : www.la-sio.fr/FootRegion/administrator

(A compléter par Alexandre)