

APPELGREN Adrien

BETHENOD Florent

RAGUIN Alexandre

SOUNDIRAM Asvina

12/06/2015 - 18/06/2015

Rapport de Projet ANDROID/JEE

GROUPE 1

EXIA - CESI

A3 RIL

LE PUY DU FOU

**Table des matières**

[Introduction 1](#_Toc422406109)

[Présentation de l’équipe 2](#_Toc422406110)

[Rappel du contexte 3](#_Toc422406111)

[Description de la Solution Proposée 4](#_Toc422406112)

[Liste des Technologies Utilisées 4](#_Toc422406113)

[Choix Technologiques 4](#_Toc422406114)

[Méthodologie & Planning des travaux 5](#_Toc422406115)

[Travail réalisé 6](#_Toc422406116)

[JEE 6](#_Toc422406117)

[Création des Entités 6](#_Toc422406118)

[Création des Services 7](#_Toc422406119)

[Création des listeners Rest 7](#_Toc422406120)

[Andoid 7](#_Toc422406121)

[Evolution possibles 8](#_Toc422406122)

[Conclusion 9](#_Toc422406123)

[Annexes 10](#_Toc422406124)

# Introduction

Dans le cadre de notre formation en troisième année à l’Exia, nous avons réalisé un projet d’une semaine inscrit dans la continuité de notre spécialité. En effet, nous sommes spécialisés en Ingénierie du Logiciel, et ce projet concerne les technologies JEE et Android. Au cours de ce rapport, nous présenterons notre équipe, puis, suite à un rappel du besoin, nous expliquerons la solution que nous proposons, les technologies utilisées ainsi que le travail accompli. Enfin, nous dresserons un bilan de ce projet et chaque membre conclura cette semaine de travail de façon personnelle.

# Présentation de l’équipe

L’équipe qui a contribué à la réussite de ce projet est constituée de quatre membres :

* Adrien Appelgren : Chef de projet
* Florent Bethenod : Développeur JEE
* Alexandre Raguin : Développeur spécialisé en technologie Android
* Asvina Soundiram : Développeur spécialisée en technologie Android

# Rappel du contexte

Le Puy du Fou, parc à thématique historique situé dans un petit village de Vendée, est l’un des quatre plus grands parcs à thème de France, et offre une multitude de spectacles pour toute la famille. En termes de vente de billet, il s’agit du deuxième parc le plus visité de notre pays. Des bénévoles retracent même l’histoire de la Vendée lors d’un spectacle nocturne, la Cinéscénie. Le parc possède d’ores et déjà une application mobile disponible sur Android.

Notre équipe a été choisie pour fournir un outil permettant à l’utilisateur de rester continuellement informé et connecté à l’application. Le but de cet outil est d’améliorer la visibilité des visiteurs concernant les horaires des programmes, le parcours personnalisé, le planning des activités etc.

Suite à la refonte annuelle de leur site web, l’application doit être fonctionnelle à 80%, posséder toutes les exigences principales, et être mise en ligne avant le 1er janvier. Les modules optionnels seront développés par la suite.

Les exigences principales correspondent à :

* Affichage du planning en temps réel
* Création du programme de la journée
* Proposition d’un planning optimisé
* Obtention des informations des activités
* Notation par le visiteur d’une activité



# Description de la Solution Proposée

## Liste des Technologies Utilisées

Le choix de l’IDE est resté libre selon les affinités de chaque utilisateur.

Une mise en commun des ressources s’est fait par l’utilisation de Git via un dépôt disponible sur GitHub.

Client : Android API 22

Serveur JEE :

* Java 8
* Maven
* Tomcat
* SPRING

Framework :

Jhipster 2.16.0

## Choix Technologiques

Nous nous sommes orientés vers une solution MVC classique, le client Android suivant le schéma standard des applications, et pour le serveur d’application, nous avons choisis une implémentation via des services REST proposés par le framework SPRING.

Pourquoi le choix de SPRING ?

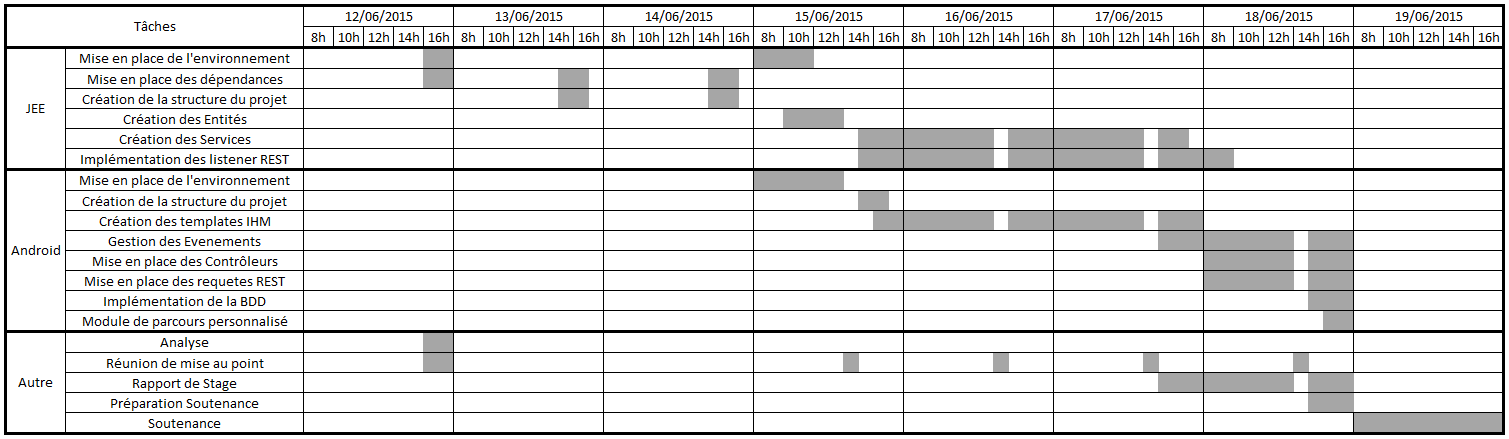
Certains membres de l’équipe était familiers des requêtes REST et pouvaient implémenter rapidement cette solution. De plus, cette solution permet de voir une manière différente de mise en place de serveur d’application.

Concernant la base de données, la version de développement est une liquibase H2 qui est stockée en mémoire sans déploiement de serveur annexe, ce qui est beaucoup plus léger et donc rapide pour le développement, cependant la mise en production nécessite un serveur MySQL fonctionnel.

L’utilisation du framework Jhipster nous a permis de mettre en place rapidement une interface web en AngularJS pour administrer les données facilement.

# Méthodologie & Planning des travaux

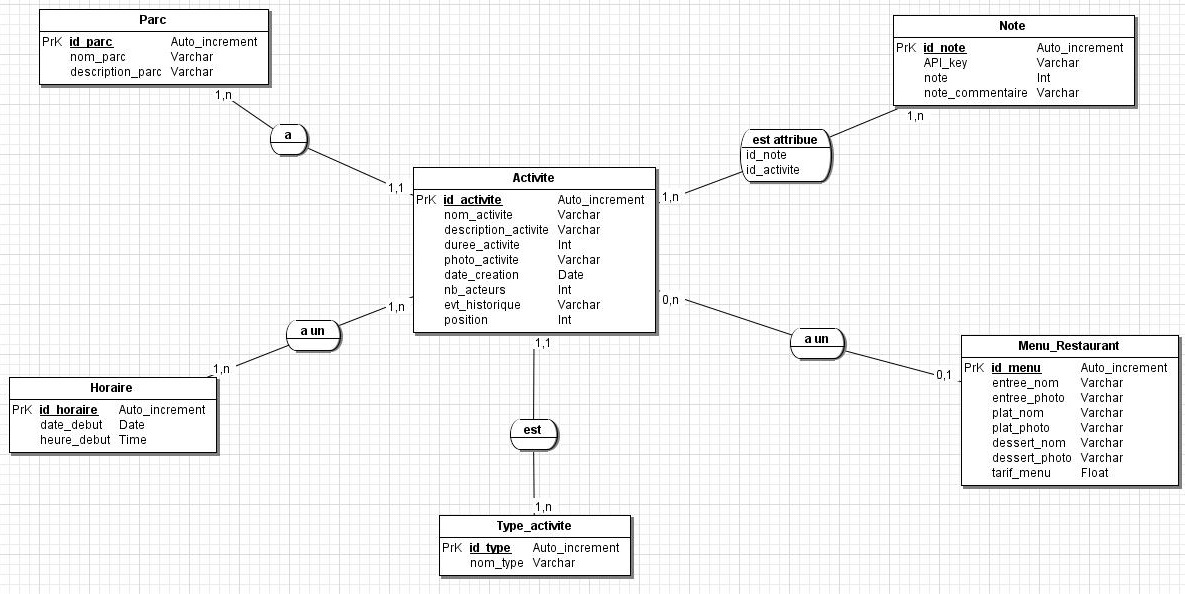
Pour ce projet, nous nous sommes basés sur une méthodologie SCRUM avec des points journaliers permettant à l’équipe de savoir l’état d’avancement du projet et l’attribution des tâches restantes.



# Travail réalisé

## JEE

### Création des Entités

La création des entités c’est construite à partir du MCD suivant :

Nous avons créé les entités via JHipster. Lors de leur création, nous avons donné un nom à l’entité puis avons renseigné ses champs ainsi que leurs types. Ensuite nous avons établis les liaisons relationnelles entre chaque entité.

Nous avons créé l’entité « Activite » qui va manager toute les autres. Elle possède une liaison many-to-many avec l’entité « Horaire »,many to one avec Parc et des liaisons one-to-many envers les autres entités.

Nous possédons donc au total 6 entités qui possèdent chacune une relation avec l’entité « Activite ».

### Création des Services

La création des services se sont faits dans le même principe que les entités, via JHipster.

Une fois la structure de celles-ci créée, nous avons implémenté leurs fonctionnalités.

Afin de répondre aux besoins, nous avons créés plusieurs services différents.

**Le PlanningService** va s’occuper de de renvoyé une liste d’activité au client. Cette liste va soit lister le planning de la journée, soit donner au client un planning optimiser afin qu’il puisse profiter un maximum du parc avant son départ.

En effet, le code de ce planning optimisé va récupérer l’heure d’arrivée du client lorsque celui-ci fais appel à ce service. De cette heure d’arrivée, il va trouver parmi toutes les activités du parc celle qui commence le plus tôt possible.

Dans ce planning sera pris en compte le temps de repas du midi ainsi que du soir mais aussi des pauses et un temps de trajet afin de se rendre à la prochaine activité.

**Le NotationService** va vérifier que si une activité a déjà été notée, celle-ci ne puisse plus l’être. Ce service permet d’éviter un abus de la part du client.

### Création des listeners Rest

L’un des concepts de Rest est la ressource. Celle-ci est un objet possédant un type, données associées, relations à d'autres ressources, et un ensemble de méthodes qui agissent sur ​​elle.

Nous avons donc créé ces ressources dans lesquels sont implémenté des méthodes standards correspondant par exemple aux méthodes GET, POST, PUT, DELETE utilisé par les objets classiques.

Pour mieux comprendre, voici quelques exemples :

Dans le listener Rest « PlanningResource » on retrouve un getter « getPlaningOpti » qui va appeler la méthode « planingOpti » dans le service associé.

Dans « NoteResource » on retrouvera des méthodes telles que la création d’une nouvelle note, la suppression d’une note ou encore l’affichage d’une note ou de son ensemble.

On peut ainsi retrouve les mêmes méthodes dans « HoraireResource » que dans celle de « NoteResource ». Celles-ci seront simplement adaptées mais la structure sera la même.

## Andoid

# Evolution possibles

# Conclusion

# Annexes