

APPELGREN Adrien

BETHENOD Florent

RAGUIN Alexandre

SOUNDIRAM Asvina

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](#_Toc422428819)

[Présentation de l’équipe 2](#_Toc422428820)

[Rappel du contexte 3](#_Toc422428821)

[Description de la Solution Proposée 4](#_Toc422428822)

[Liste des Technologies Utilisées 4](#_Toc422428823)

[Choix Technologiques 4](#_Toc422428824)

[Méthodologie & Planning des travaux 5](#_Toc422428825)

[Travail réalisé 7](#_Toc422428826)

[JEE 7](#_Toc422428827)

[Création des Entités 7](#_Toc422428828)

[Création des Services 8](#_Toc422428829)

[Création des listeners Rest 8](#_Toc422428830)

[Android 9](#_Toc422428831)

[Evolution possibles 13](#_Toc422428832)

[Conclusion 14](#_Toc422428833)

# 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. L’application Android doit obligatoirement passer par des web services, reliés à une base de données.

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, même si pour le développement Android nous avons opté pour Android Studio, grâce à sa facilité de mise en place.

Une mise en commun des ressources a été régulièrement réalisée par l’utilisation du logiciel de sauvegarde et de gestion de version Git, via un dépôt disponible sur GitHub. Nous avons préféré Git à d’autres logiciels comme Subversion car nous avions tous l’habitude de l’utiliser pour nos projets personnels.

Au niveau du client :

Android API 22

Au niveau du serveur JEE  nous avons utilisé de nombreuses technologies:

* Java 8
* Maven
* Tomcat
* SPRING

## Choix Technologiques

Nous nous sommes orientés vers une solution MVC classique, le client Android suit le schéma standard des applications. Au niveau du serveur d’application, nous avons choisi une implémentation via des services REST proposés par le Framework SPRING.

Pourquoi le choix de SPRING ?

Certains membres de l’équipe connaissaient déjà ce Framework et manipulaient avec aisance les requêtes REST. Ce choix fût donc une évidence puisqu’il était plus rapide pour nous d’implémenter cette solution. De plus, cela nous a permis d’apprendre une manière différente de mettre en place un serveur d’application.

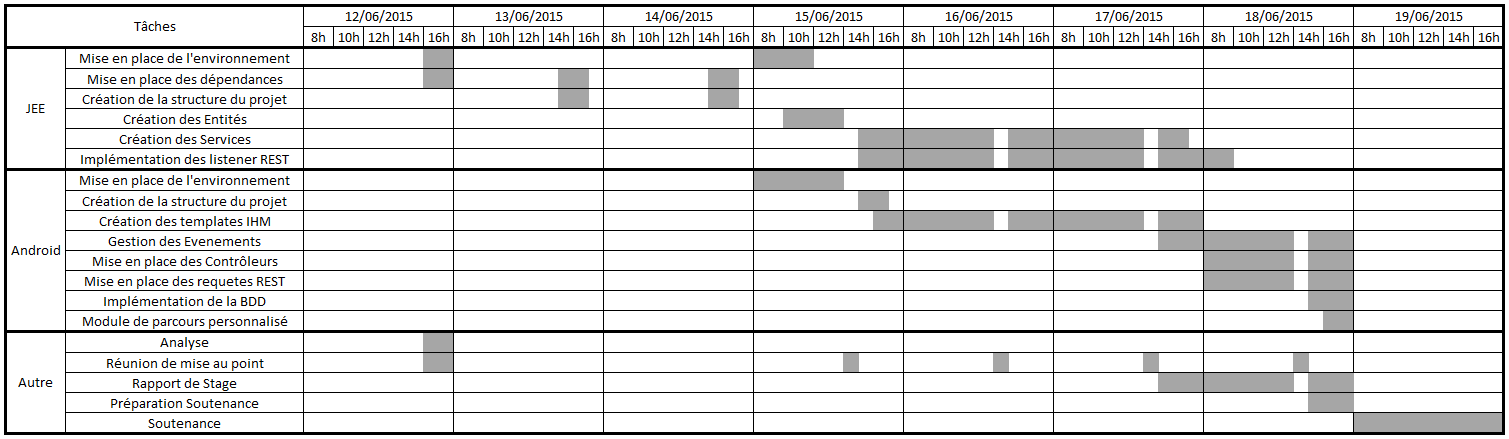
Concernant la base de données, la version de développement est une liquibase H2 qui est stockée en mémoire sans déploiement de serveur annexe, ce qui est beaucoup plus léger et donc plus rapide pour le développement, cependant la mise en production nécessite un serveur MySQL fonctionnel.

# Méthodologie & Planning des travaux

Pour ce projet, nous nous sommes basés sur une méthodologie SCRUM, aussi considérée comme une méthode AGILE. Cette méthode de gestion de projet permet de répondre à des problèmes complexes et changeants en réalisant des sprints, période assez courte durant laquelle l’équipe rend un livrable potentiellement terminé au client.

Au début du projet, nous avons désigné un chef de projet, puis nous avons analysé la demande du client. Le chef de projet a ainsi proposé des technologies à utiliser, et ensemble, nous avons réalisé un MCD pour la base de données. Une fois que la méthodologie adoptée a été validée par tous, le chef de projet a défini nos rôles durant cette semaine. Suite à cette répartition, il nous a également donné des tâches à réaliser avant la fin de la journée. De ce fait, tous les jours, nous avions des jalons d’une journée ou d’une demi-journée à respecter, pour le bon déroulement du projet. Malgré quelques problèmes liés à la stabilité de l’environnement, nous avons su réaliser nos tâches administrées en temps et en heures.

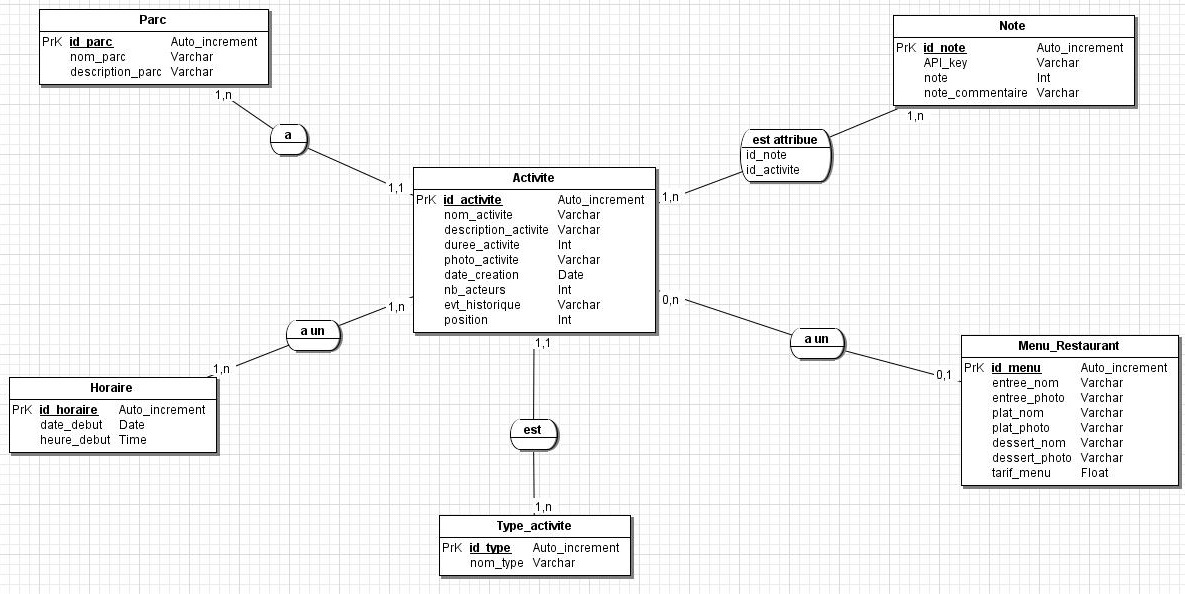
Chaque jour, nous présentions notre travail aux autres membres de l’équipe durant les stands-up, ou mêlées quotidienne, qui sont des réunions quotidiennes lors desquelles nous montrons aux autres membres ce que nous avons réalisé, les problèmes rencontrés et ce qu’il nous reste à faire. Ceux-ci permettent à l’équipe de se situer en fonction de l’état d’avancement du projet, de participer à d’autres tâches et d’aider les autres membres de l’équipe.

Voici le planning représentant le déroulement de nos sprints durant la semaine:

# Travail réalisé

## JEE

### Création des Entités

La création des entités de notre base de données s’est construite à partir du MCD suivant :

Nous avons créé les entités en utilisant le Framework JHipster. Lors de sa création, nous avons donné un nom à l’entité puis avons renseigné leurs champs ainsi que leurs types. Ensuite nous avons établi les liaisons relationnelles entre les entités.

Nous avons créé l’entité « Activite » qui va manager toutes les autres. Elle possède une liaison many-to-many avec l’entité « Horaire », many-to-one avec « Parc » et des liaisons one-to-many avec les autres entités. Nous possédons au total 6 entités qui nous permettront de réaliser des échanges d’informations entre la base de données et l’application.

### Création des Services

La création des services s’est faite dans le même principe que les entités, grâce à JHipster qui propose de nombreux outils adaptés à notre besoin.

Une fois la structure de ces services créée, nous avons implémenté leurs fonctionnalités. Bien sûr, afin de répondre aux besoins de notre client, nous avons créés plusieurs services différents :

* **Le PlanningService** va s’occuper 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 web service de ce planning optimisé va récupérer l’heure d’arrivée du client lorsque celui-ci fais appel à ce service. A partir de celle-ci, le service 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, des données associées, des relations à d'autres ressources, et un ensemble de méthodes qui agissent sur ​​elle.

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

Pour mieux comprendre, voici quelques exemples :

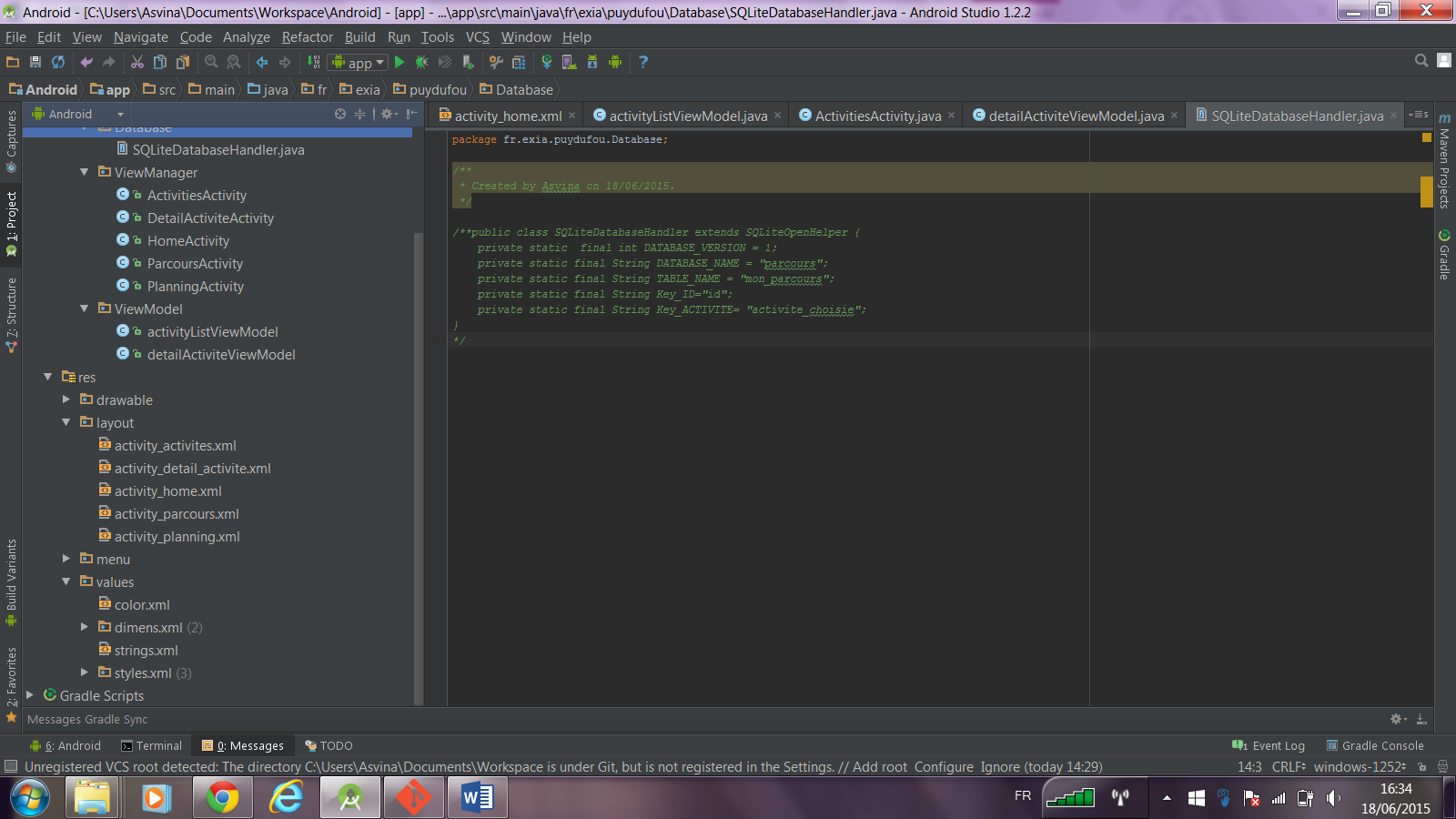
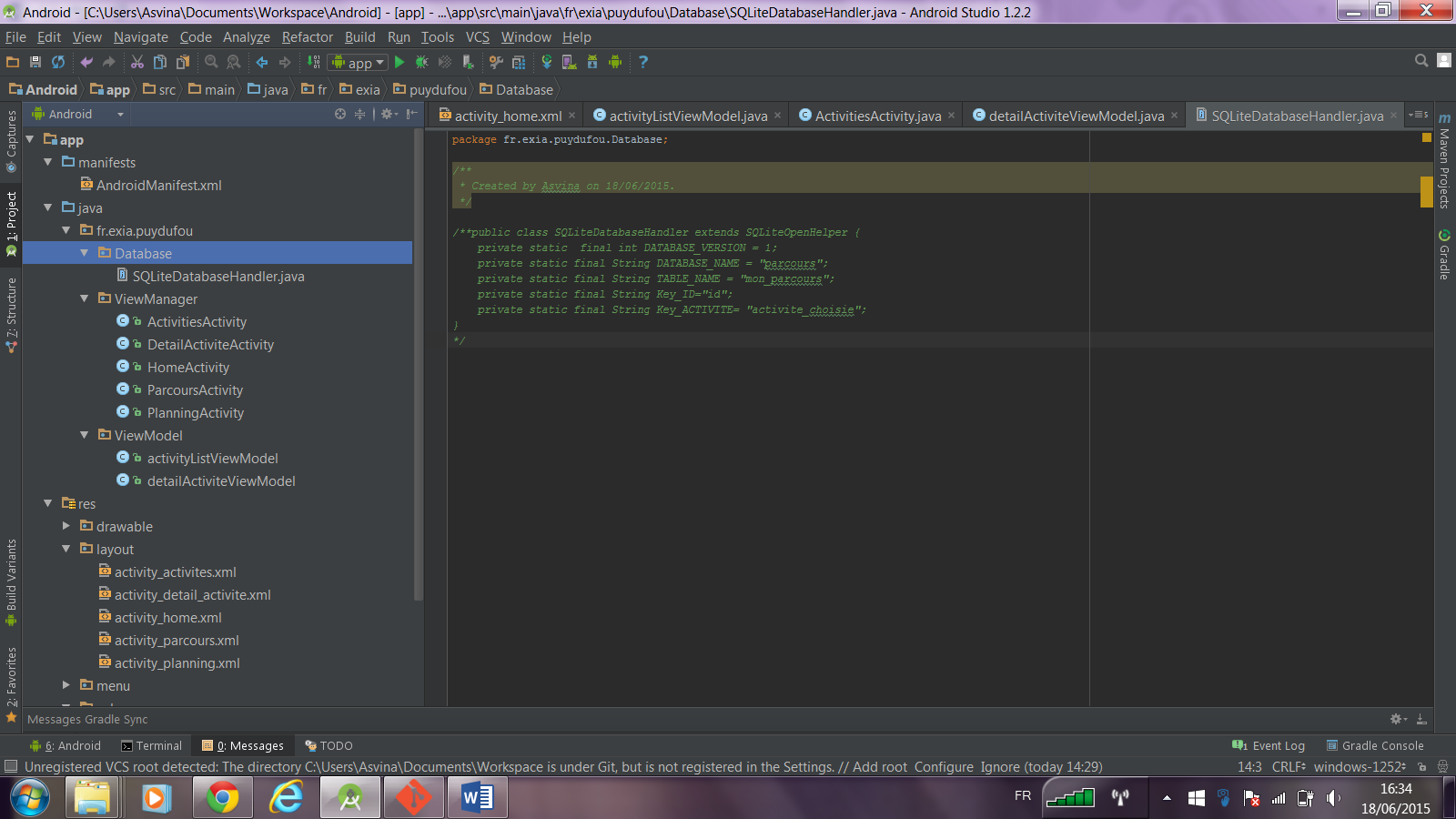
Dans le listener Rest « PlanningResource » on retrouve un getter « getPlaningOpti » qui va appeler la méthode « planingOpti » dans le service associé.

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 retrouver les mêmes méthodes dans « HoraireResource » et dans celle de « NoteResource ». Celles-ci seront simplement adaptées selon la classe, mais la structure sera la même.

## Android

Voici l’arborescence de notre application Android :



Au niveau de la structure de notre projet Android, nous avons plusieurs répertoires automatiquement crées lors de la création d’un projet Android, nous avons par la suite ajouté nos propres packages, notre base de données interne, nos fichiers XML, nos classes ainsi que nos vues et feuilles de style. Ci-dessous, une explication détaillée de chaque dossiers/fichiers par rapport à ce qu’il apporte à notre application Android.

* Manifest : qui contient le fichier AndroidManifest.xml. Ce fichier permet de gérer les autorisations de l’application Android ainsi que de déclarer les activités.
* Java : dossier contenant les packages de notre application. Le package *fr.exia.puydufou* regroupe trois autres packages, Database, ViewManager et ViewModel. Ces packages contiennent toutes les classes Java de notre application.

Le package Database contient un fichier SQLiteDatabaseHandler.java, qui permet de créer une base de données SQLite interne. Cette base de données permet de gérer des requêtes légères qui n’ont pas besoin de passer par le serveur pour être exécutées. Ce fichier contient la création des tables de la base de données, les champs ainsi que la définition des types et les méthodes qui seront utilisées pour récupérer les informations et les afficher sur le téléphone du client.

Le package ViewManager possède les classes reliées aux vues de notre application.

Le package ViewModel contient les méthodes utilisées dans nos classes.

Res : Il s’agit du dossier contenant les ressources de l’application, c’est-à-dire :

- les images (dossier drawable) avec les icones, logo de l’application, image de fond etc.

- les vues (dossier layout) avec les activités de notre application, autrement dit les templates de chaque page.

- les values, qui sont des fichiers permettant de gérer toute la partie graphique des activités, c’est-à-dire les couleurs, les textes, les dimensions etc. Le fichier de style est également enregistré dans ce dossier.

- le menu définit simplement quelle sera la page d’accueil de l’application.

* Gradle Scripts : Ce sont des scripts de configurations, pour définir les propriétés de compilation, et notamment pour définir les propriétés locales (le chemin du sdk).

Dans notre application, nous avons cinq classes, reliées à cinq vues, et qui sont les différentes activités :

La première activité est la page d’accueil, elle contient trois boutons. Le premier renvoit vers la liste de toutes les activités, le second affiche le parcours personnalisé du client pour la journée en cours, et le dernier lui permet d’afficher son planning.

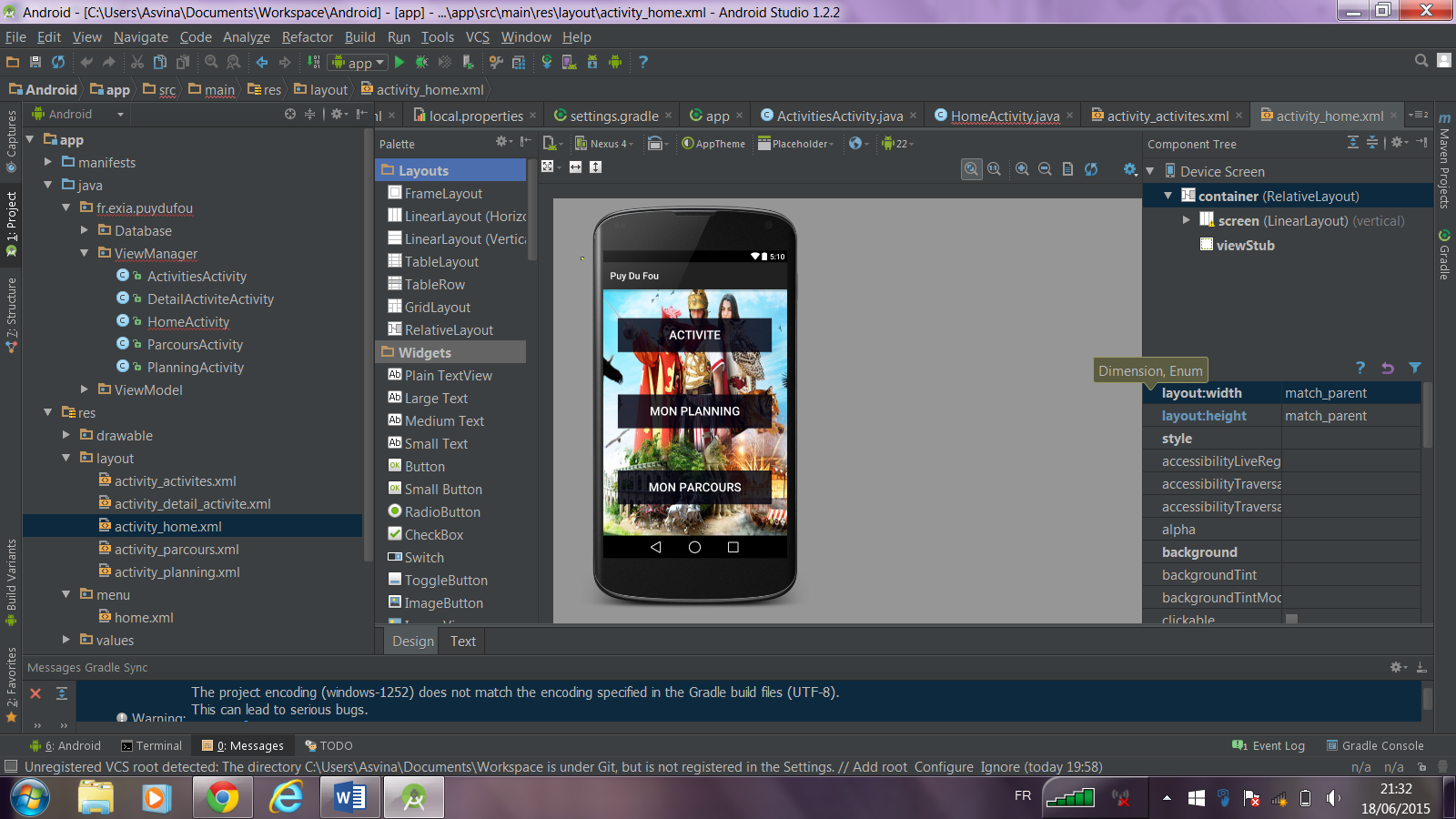
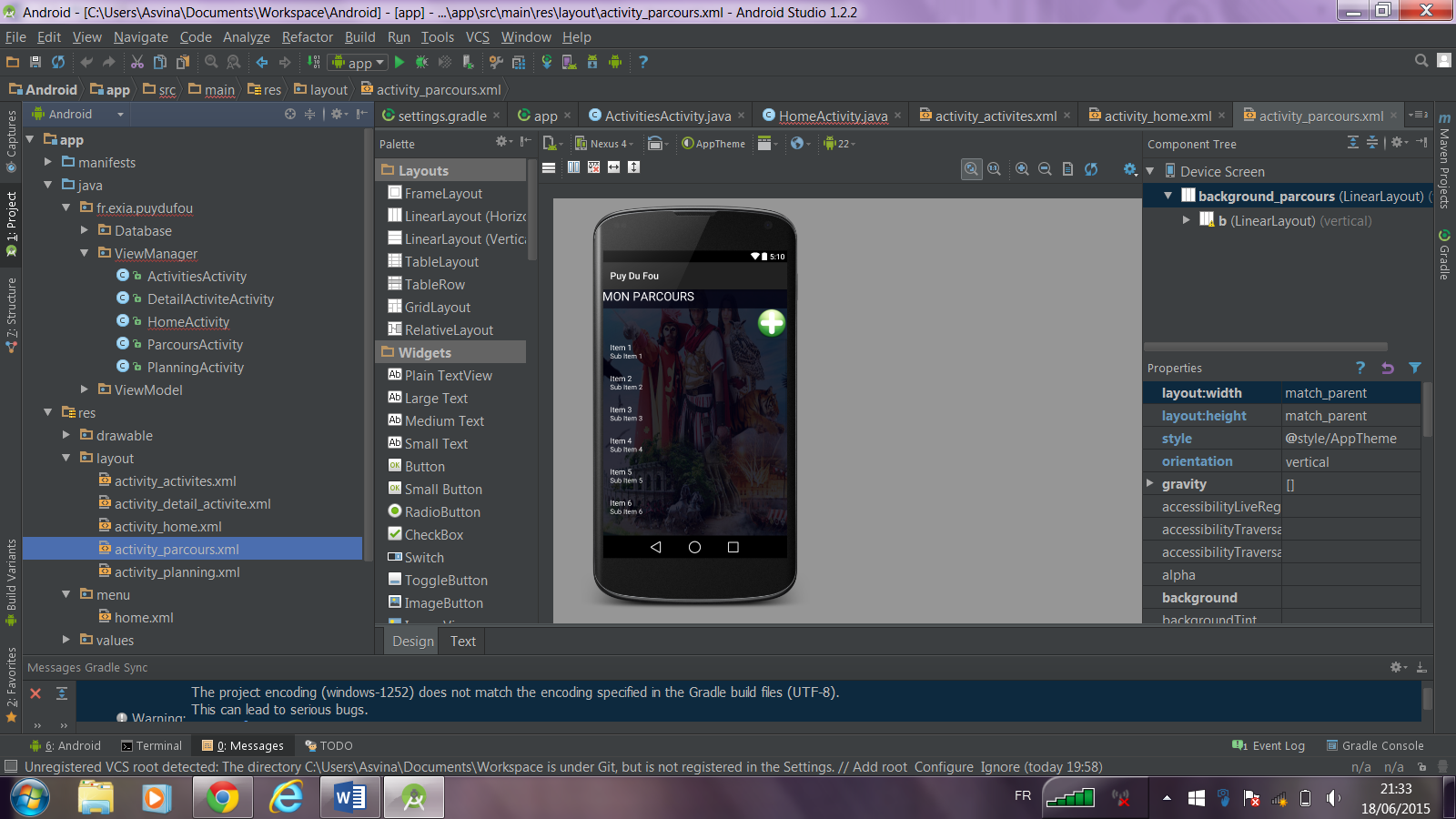
La seconde activité est la liste des activités. Elle permet au visiteur d’avoir sur une même page, l’ensemble des spectacles disponibles. Elle est reliée au détail de chaque activité.

La troisième activité est le planning, il permet d’afficher un planning de la journée, optimisé en fonction des heures et durée des spectacles, et des activités choisies dans le parcours.

La quatrième activité est le parcours, qui consiste à afficher les activités que l’utilisateur à choisi. Sont indiqués les horaires, les noms des activités, les durées et leur position dans le parc.

La dernière activité est le détail d’une activité, qui s’ouvre lors du clic sur une activité de la liste. Elle permet au visiteur d’avoir des informations sur toutes les activités du parc (heure, durée, acteurs, date de création, description etc).

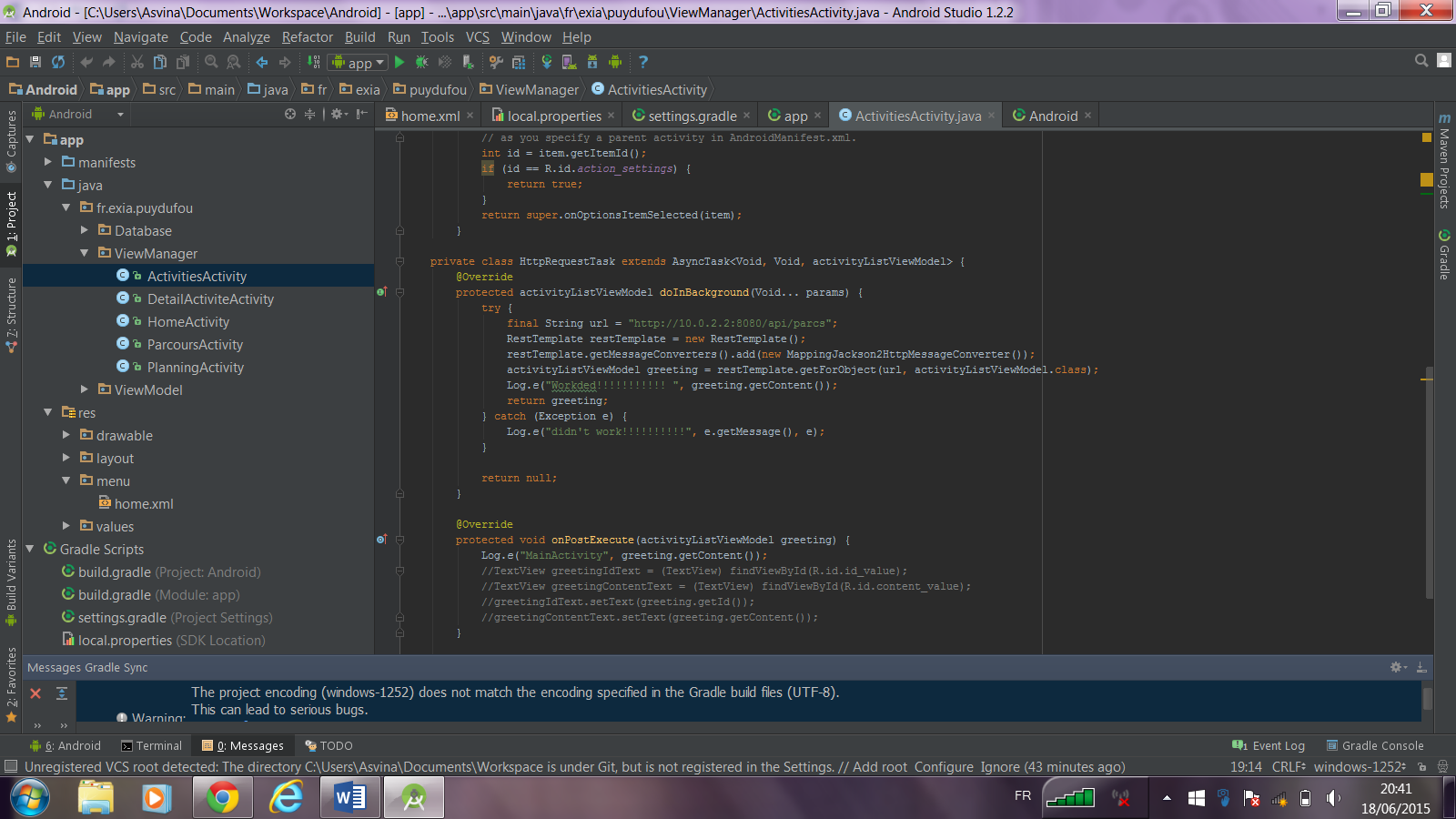
Voici deux vues de notre application Android :

Activité Accueil Activité Parcours

Notre application Android est reliée aux web services par des requêtes REST, grâce au Framework Spring. Pour cela il a fallu ajouter une autorisation dans le fichier AndroidManifest.XML. Il suffit ensuite de réaliser une requête REST depuis la classe HttpRequestTask qui est étendue à AsyncTask. Nous avons ainsi ajouté une méthode qui va être réalisée en arrière-plan et de façon asynchrone, pour optimiser l’application, sa fluidité et sa rapidité.

Voici le code de la méthode implémentée pour notre requête de façon asynchrone :



# Evolution possibles

# Conclusion

Conclusion personnelle Asvina :

Ce projet est pour moi l’accomplissement de nombreuses heures de travail. Il se trouve que j’ai toujours eu une certaine appréhension pour le développement en java, car je n’avais jusque-là, encore jamais réalisé de véritables applications dans ce langage. Il a été difficile pour moi de suivre l’Unité d’Enseignement JEE, et j’étais perdue assez facilement. Pourtant, depuis le début de ce projet, le fait d’appliquer mes connaissances sur un sujet réel, et dans un véritable projet en groupe, m’a permis de comprendre de nombreux points qui m’étaient encore flous, notamment le lien entre le JEE et la partie Android. Je pense tenter à nouveau de coder des applications Android car les technologies embarquées sont très demandées de nos jours, et j’ai également bien apprécié cette partie de’ l’UE.

Pour ma part, je suis ravie d’avoir participé à ce projet, car nous avons réussi à trouver une certaine cohésion dans notre travail. Le petit plus de cette expérience aura été la répartition efficace des tâches avec le partage des documents via l’outil GIT. Ce logiciel nous a permis de gagner énormément de temps, de sorte à ce que l’on puisse tous accéder aux ressources des autres. Et c’est exactement le but d’un projet de ce genre, apprendre à travailler ensemble, et trouver les bon outils pour avancer le plus efficacement possible. Il se trouve que notre méthode de travail a été remarquablement bien structurée, et tous les membres de l’équipe ont contribué à cette réussite.