

# Université de La Rochelle



Master 1 ICONE  
Rapport du projet

---

## **ECO-LEAVES** **Réseau social associatif environnemental**

---



Réalisé par : Thomas Pepper-Mohammed Benaou-Christian Marquay-Zakaria El  
hajjami-Thomas Chambon-Ibrahima Guiro  
Enseignant Référent: M. Demko  
Encadré par : M. Julien Perez

# 1.Introduction

## 1.1 Presentation du sujet

Dans le cadre, de notre premier semestre de Master au sein de l'université de La Rochelle, nous avons à faire un projet transversale regroupant 4 unités d'enseignements (Ingénierie logicielle, AOSI, techniques de programmation et Base de donnée avancée). Pour ce projet, il y avait plusieurs sujets et nous avons été tiré au sort pour travailler sur le Sujet de l'entreprise Cat-Amania.

Le sujet est la création d'un RSE ( Réseau social d'entreprise ) sous forme d'un application mobile mais pour une association vu que nous il s'agit d'un projet sans budget.

Le RSE devrait avoir plusieurs fonctionnalités qui sont eux, libre de notre choix. Ceci dit, il y avait des technologies imposés pour effectuer ce projet, par exemple, il fallait avoir un Back-end qui fonctionne sous Java, un architecture Microservices et un application cross plateforme (Android et IOS). Puis environ 30% en commun avec le groupe numéro 2, qui travaille sur le même sujet que nous.

## 1.2 Presentation de la solution

La plateforme ECO-LEAVES a pour objectif de permettre aux utilisateurs de se rencontrer en ligne afin d'organiser des événements associatifs environnementaux et d'échanger des objets qui ne leur sont plus utiles. Cet échange a pour but de combattre l'obsolescence programmée et le gaspillage des ressources.

Les utilisateurs de la plateforme sont des citoyens, membres ou non d'associations sensibilisées aux enjeux environnementaux et de développement durable.

Le Core fonctionne sous JAVA, langage indépendant de tout processeur et de tout système d'exploitation avec un fort potentiel d'industrialisation.

Les bases de données fonctionnent sous Elasticsearch, un serveur utilisant Lucene pour l'indexation et la recherche des données. Il fournit un moteur de recherche distribué et multi-entité à travers une interface REST. L'échange des données se fait au format JSON.

L'application mobile fonctionne sous Cordova, permettant de créer des applications pour différentes plateformes en HTML, CSS et JavaScript . La partie JavaScript contient l'application de messagerie (chat) pour mettre en contact les utilisateurs de la plateforme.

L'application sera déployée sur les plateformes IOS et Android.

## 2. Gestion du projet

Nous avons réalisé une répartition des tâches pour chaque membre de l'équipe une tâche(s) à faire, à travers l'outil "Trello", ce dernier est un outil qui nous permet de gérer tous les aspects de la gestion de projet.



### 2.1 Outils de l'équipe et découpage des tâches

Afin de pouvoir travailler en équipe efficacement, nous utilisons l'outil GitLab. Cela nous permet de partager notre travail, de le récupérer instantanément, et de travailler en parallèle sur plusieurs tâches.



### 2.2 Mise en place de l'équipe

Notre équipe est constituée de 6 personnes, cette dernière devait suivre le schéma-type d'une équipe utilisant la méthode Scrum :

- Un chef de projet, qui devra organiser l'équipe, communiquer avec le client..etc.

Notre chef de projet est Thomas PEPPER

- Un chef de projet, qui devra vérifier qu'il n'y a pas d'incohérence entre tous les différents tâches réalisés.

Notre chef de projet est Christian MARQUAY

- Quatre développeurs qui travaillent seul ou en équipe sur différentes tâches qui leur seront attribuées.

Nos développeurs sont Mohammed BENAOU, Thomas CHAMBON, Ibrahima GUIRO, Zakaria EL HAJJAMI .

Des réunions matin et soir ont été organisées entre tous les membres, pour organiser le travail, discuter des points sur lequel se concentrer, et faire le point sur le travail effectué et le travail à faire.

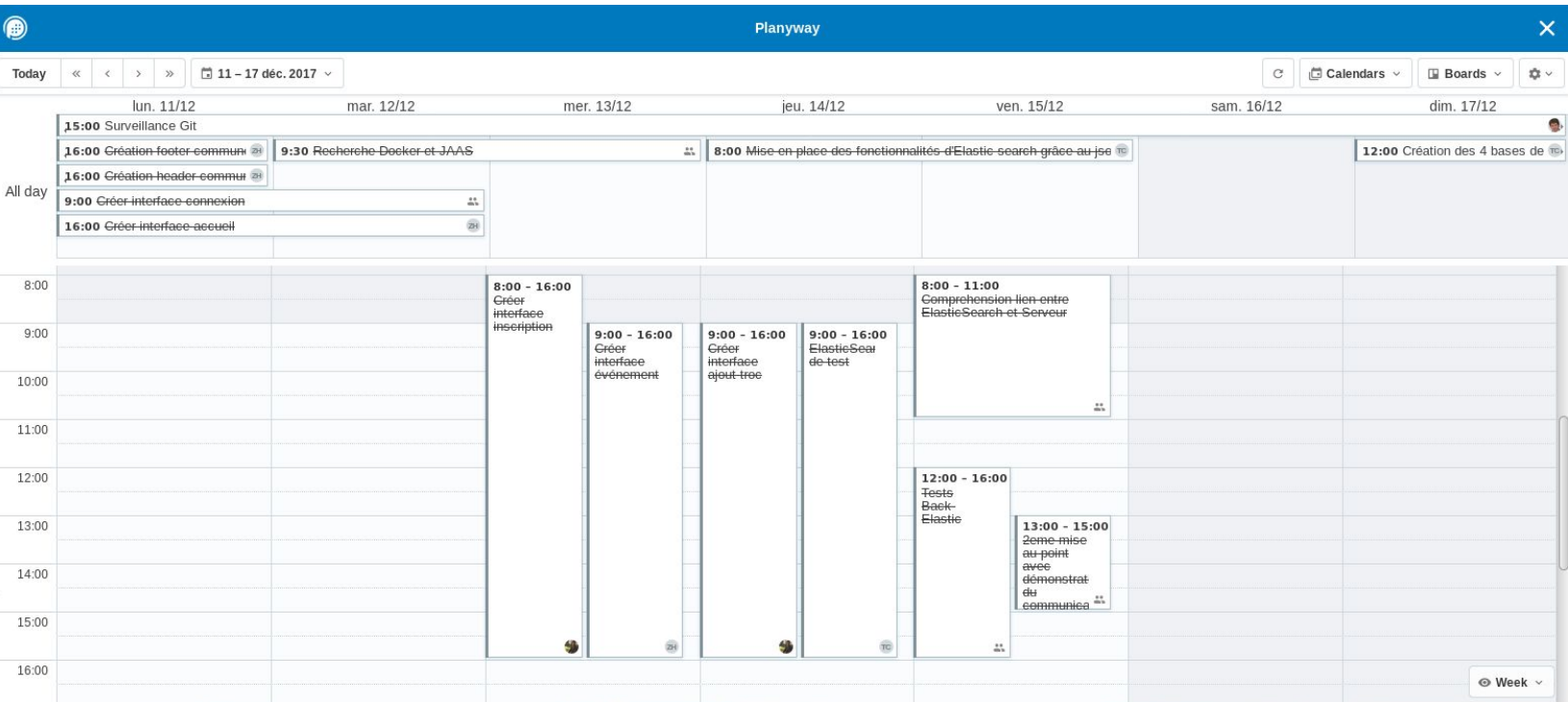
## 2.3 La methode Agile

Durant tout le long de ce projet nous avons essayé de suivre la méthode Agile, une méthode de Gestion de projet vu en Ingénierie Logicielle qui consiste à avoir en fin de chaque Sprint, quelque chose de fonctionnel. Notre Client M. Perez nous a déconseillé à utiliser des Sprints vu la durée du projet mais nous avons quand même mis en place des pseudo-sprints, chaque fin de semaine correspondent à la fin d'un Sprint.

## 2.4 Planning

Pour effectuer le planning et pour que M. Perez puisse le visualiser à tout moment, nous avons décidé de prendre un plugin directement intégré à Trello : "Planway Calender". Ce plugin permet de mettre en place un planning avec les tâches prescrit dans notre Trello et les répartir pour chaque membre de l'équipe.

Voici un exemple de planning pour la semaine 50 (Les tâches rayés sont les tâches accomplies).



## 3. Choix technologiques

### 3.1 Elasticsearch

Elasticsearch est un moteur de recherche sur tout type de document, ayant pour avantages :

- Architecture adaptable (montée en charge).
- Utilise le format JSON via des requêtes HTTP (moteur de recherche utilisable avec n'importe quel langage de programmation).
- Interactions avec la base par requête REST.
- Open Source

### 3.2 Java / JEE

Java est un langage de programmation orienté objet. Ce langage a été imposé pour le back-end de l'application, il a notamment pour avantages :

- Langage facilement portable (JVM).
- Résultats indépendants de la plateforme d'exécution.
- Applications facilement industrialisables (JEE).

### 3.3 Cordova

Framework de développement sur différentes plateformes notamment mobiles (iOS/Android). Ce framework repose sur les trois langages web HTML, JS et CSS et est open source. Il offre une portabilité accrue du code (iOS/Android) et une facilité de développement des langages de programmation web "simples" (HTML/CSS).

### 3.4 JQuery / ChatJS

ChatJS est un plugin jQuery pour la messagerie instantanée, qui est côté client. La communication côté serveur est implémentée via des adaptateurs. L'adaptateur est le morceau de code qui est responsable de la gestion de la communication client / serveur.

### 3.5 Sécurité

Différents aspects de la sécurité doivent être implémentés lors de la programmation de l'application mobile : authentication, autorisations, cryptage, sessions, cookies, etc. La sécurité doit être incluse des deux côtés : client et serveur. L'authentification permet d'identifier un utilisateur, ne serait-ce que par son login et son mot de passe. Pour l'authentification nous allons utiliser JAAS (Java Authentication and authorization service), un framework de sécurité de Java. Il permet l'authentification des utilisateurs utilisées dans une application Java. Une API Key est une clé cryptée qui permet d'identifier qui appelle le programme - un utilisateur ou un développeur - pour avoir des logs d'utilisation de la plateforme. Avec Cordova, on peut ajouter une liste blanche qui contrôle l'accès à des domaines externes sur lesquels l'application n'a aucun contrôle. Concernant les bases de données, les mots de passe doivent au minimum être cryptés en sha-1.

### 3.6 Mode de déploiement

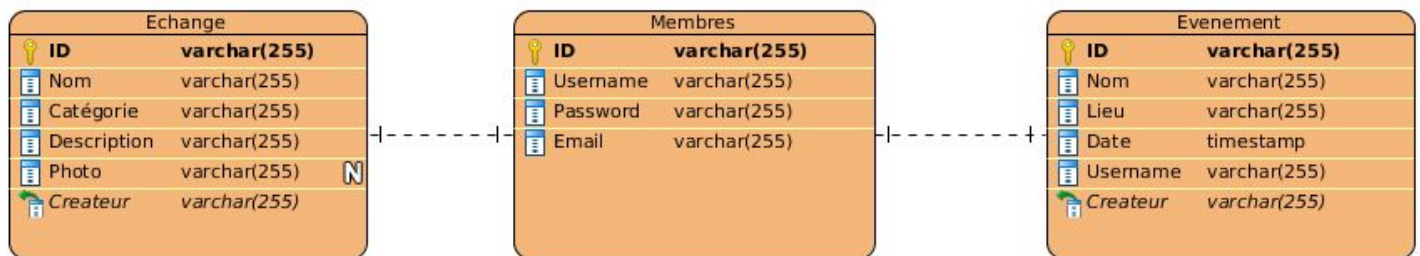
Pour déployer notre application sur le cloud, nous avons utilisé le framework Docker pour la virtualisation d'infrastructure . Il fournit une structure spéciale appelée container, légère et isolée des machines. Cela permet d'éviter de configurer chaque serveur, et d'envoyer juste les conteneurs dont on a besoin, en étant certain qu'ils marchent partout pareil, peu importe la configuration à l'arrivée. On virtualise des services, qu'on combine, et non une machine complète. Docker peut donner une meilleure expérience au développement d'applications, surtout quand cela concerne des applications large-scale avec des environnements complexes et un temps de compilation long. Docker peut également aider à fournir un meilleur environnement de test et de build, et à allouer les ressources plus efficacement. Par la suite pour déployer l'application, nous avons mis nos "containers" docker sur une plateforme cloud: "Heroku". Heroku permet le déploiement très rapide d'applications web dans le cloud, avec une gestion très souple au travers d'un modèle de gestion des processus emprunté à Unix et adapté au Web .

## 4. Architecture

### 4.1 Base de données

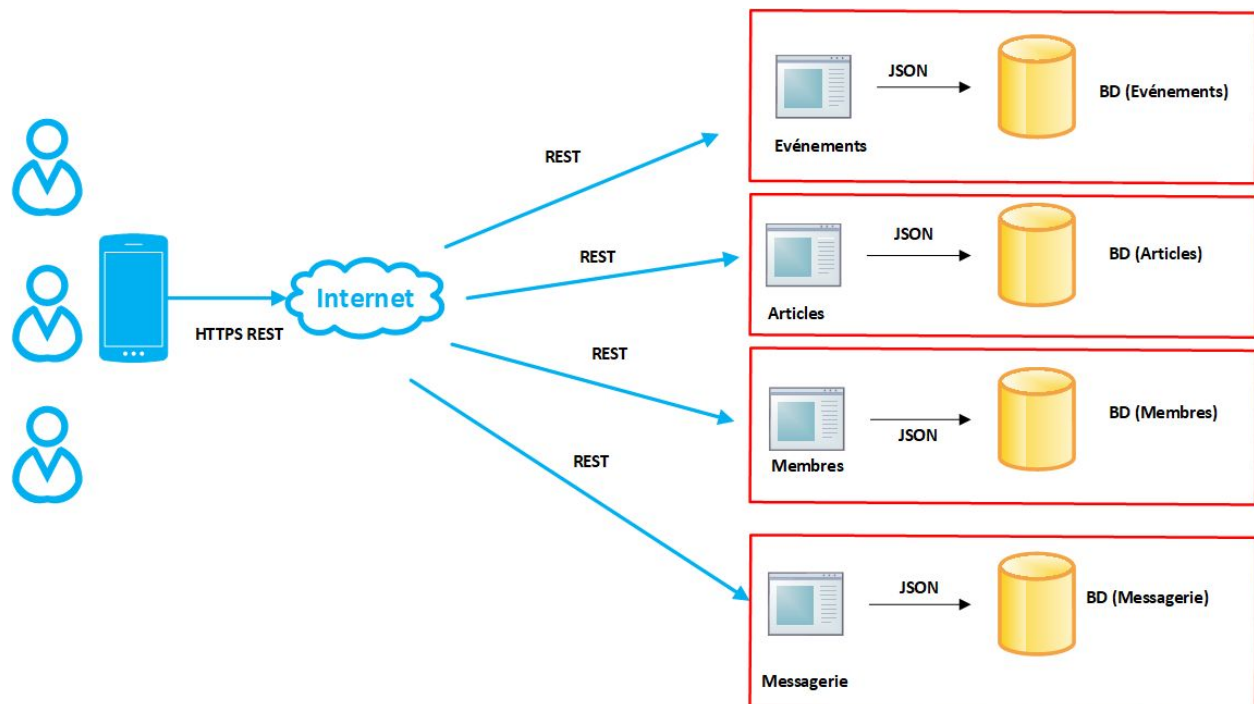
Pour nos base de données nous sommes parties sur le choix d'utiliser elasticsearch pour faciliter l'utilisation des fonctions de recherche. Au départ nous étions parti sur une base de 4 base de donnée différents : Membres, Chat, Événements et Échanges.

Chaque base correspond à chaque microservice de notre application, notre base membre contient l'id, username, password et mail de chaque membre. La base Evenement contient l'id, le nom, description, la date et le lieu de l'événement. C'est la meme chose pour Échange, il contient les éléments de base l'id, le nom, la description et une photo. Chaque événement et chaque échange à un seul propriétaire, le membre qui l'as créé. Le but de ses bases est de pouvoir créer des événements et des échanges pour que tous les autres membres sur l'application puisse voir.



Notre schéma de base de donnée

## 4.2 Les microservices



*Architecture réseau*



## 5. Le déroulement

### 5.1 Première semaine

Notre projet a débuté le 06/11/2017 par une réunion avec les professeurs et les intervenants proposant les projets. Nous avons été tiré au sort pour travailler sur le projet de Cat-Amania sur le développement d'un RSE pour une association. Notre matinée a été réservée pour l'explication et nos questions sur le projet à venir. Le but de notre première aux yeux de notre Client (M. Perez) était de mettre en place le dossier de conception, donc un dossier comprenant nos idées et notre architecture pour l'application.

Par la suite, en tant qu'un groupe, nous avons fait un "BrainStorming" pour mettre en commun tout nos idées et pouvoir préparer le dossier de conception comme il le fallait.

Thomas Chambon et Christian ont été désigné pour travailler sur le dossier de conception en son intégralité, Mohammed et Zakariae ont eux de leur côté travaillé sur une maquette du Front-End de l'application. Thomas Pepper quant à lui a travaillé sur la gestion du projet comme par exemple toutes les tâches à effectuer ainsi que le planning du projet. La dernière grande tâche de cette première semaine du projet a été les recherches sur les Microservices et le déploiement, cette tâche a été effectuée par Ibrahima.

Durant cette semaine, nous avons rencontrés très peu de problèmes, la seule problème qui s'est opposé à nous c'est le choix des technologies. Certains membres du groupe ont souhaité partir sur d'autres choix, des technologies avec lesquels nous avons plus de facilités suite à nos cours de Licence.

Pour la présentation de cette fin de semaine, nous avons du coup montré notre dossier de conception ainsi que les maquettes faites sous Photoshop.

## 5.2 Deuxième semaine

En suivant notre planning et notre méthode Agile, en fin de cette deuxième semaine, l'objectif était d'avoir un front et une base de donnée à montrer.

La répartition des tâches s'est effectué de la manière suivante. Thomas Chambon doté de compétences en base de données a donc été choisi pour travailler sur la partie Elasticsearch en collaboration avec Gregory Daly (membre de l'équipe 1), la partie que nous avons mis en commun avec l'autre équipe. Mohammed et Zakariae ont continué de travailler ensemble mais cette fois ci sur les différents pages HTML/CSS/JS qui compose notre Front-End que nous allons transformé en application Android/IOS plus tard dans le projet. Christian lui a travaillé sur la mise en place d'un serveur en Java et aussi en collaboration avec Ibrahima sur les recherches sur le déploiement donc Docker et le plateforme Cloud. Toute la partie Gestion de Projet a été effectué par Thomas Pepper ainsi que les diaporamas et les documentation.

Cette semaine a été très mouvementés suite à des problèmes des communication avec M. Perez, notre réunion avec lui, a dû être annulé à cause d'un imprévu client. Alors pour nos questions et pour nous rassurer sur nos choix, nous avons pu communiquer seulement par mail. L'autre problème rencontré durant cette pseudo-sprint était dû à la faites qu'on était dépendant de l'avancement d'elasticsearch, nous avons préféré attendre avant d'attaquer le back-end et ne pas coder à l'aveugle pour éviter des problemes eventuels.

En fin de semaine, les objectifs ont été remplis dans la plupart, nous avons eu un Elasticsearch qui fonctionnait (quelques fonctionnalités toujours pas implémentés) et un Font-End opérationnel et responsive.

### 5.3 Dernière semaine

Pour la dernière semaine de ce projet transversale, nous avons prévu d'effectuer l'intégration et les liens de nos "briques" soit disant le Front, le serveur et les base de données.

Les objectifs pour cette semaine n'ont pas pu être accompli à cause d'un manque de temps, nous avons eu énormément de soucis pour relier toutes les briques et les faire fonctionner ensemble. Le plus gros problème a été la mise en place de notre serveur sous forme

## 6. Analyse

Les principales fonctionnalités manquantes qui sont present dans notre dossier de conception sont: le système de messagerie instantanée, le déploiement dans docker et encore sous un plateforme Cloud.

## 6. Conclusion

Au final, notre application ne fonctionne pas comme on l'aurait souhaité quand nous avons mis en place notre dossier de conception. Il y a fonctionnalités qui n'ont jamais abouti suite à un manque de temps.

L'ensemble du groupe a trouvé dans l'ensemble ce projet très enrichissante malgré les nombreuses difficultés rencontrés durant le projet. Ça a été une première expérience dans ce genre projet à travers une période de 3 semaines. Un projet où il a fallu mettre en oeuvre en même temps nos compétences niveau gestion d'un projet mais aussi niveau connaissances théorique et pratique.

Le travail en équipe a été primordial tout au long de ces 3 semaines, on a dû faire confiance aux capacités des uns comme les autres. Cependant que ça a été un projet en groupe, nous avons dû effectué la plupart de nos tâches seul pour que le groupe puisse avancer rapidement en bloc.

On a aussi pu voir les conditions d'être en entreprise grâce aux faites d'avoir un deadline et un vrai client derrière nous le long du projet, d'avoir ce client derrière nous, nous a poussé de donner le maximum de nous pour pouvoir rendre le projet en temps et en heure.

Si nous pourrions revenir en arrière et recommencer ce projet, nous aurions changer nos de technologie non à cause de la complexité de nos choix mais à cause du deadline. Nous avons vu ambitieux dès le départ, peut-être trop ambitieux pour un projet sur 3 semaines.

Le sujet en lui même du projet était vraiment pertinent pour des étudiants en Master et nous avons tous pris énormément de plaisir à travailler sur ce projet malgré la finalité du projet. Les seules choses qu'on reproche au niveau du sujet est le faite qu'il soit trop vaste et qu'il a énormément de solutions possibles et le faite qu'on soit partie de 0 et qu'il fallait tout faire.

## 7. Table des matières

<b>1.Introduction</b>	<b>2</b>
1.1 Presentation du sujet	2
1.2 Presentation de la solution	2
1.3 Architecture Reseau	2
<b>2. Gestion du projet</b>	<b>4</b>
2.1 Outils de l'équipe et découpage des tâches	4
2.2 Mise en place de l'equipe	4
2.3 La methode Agile	5
2.4 Planning	6
<b>3. Choix technologiques</b>	<b>7</b>
3.1 Elasticsearch	7
3.2 Java / JEE	7
3.3 Cordova	7
3.4 JQuery / ChatJS	7
3.5 Sécurité	7
3.6 Mode de déploiement	8
<b>4. Architecture</b>	<b>9</b>
4.1 Base de données	9
<b>Notre schéma de base de donnée</b>	<b>9</b>
4.2 Les microservices	10
<b>5. Le deroulement</b>	<b>10</b>
5.1 Première semaine	10
5.2 Deuxième semaine	11
5.3 Dernière semaine	12
<b>6. Conclusion</b>	<b>13</b>
<b>7. Table des matières</b>	<b>14</b>
<b>Annexes</b>	<b>15</b>
<b>Recherches / Sources</b>	<b>18</b>

## Annexes

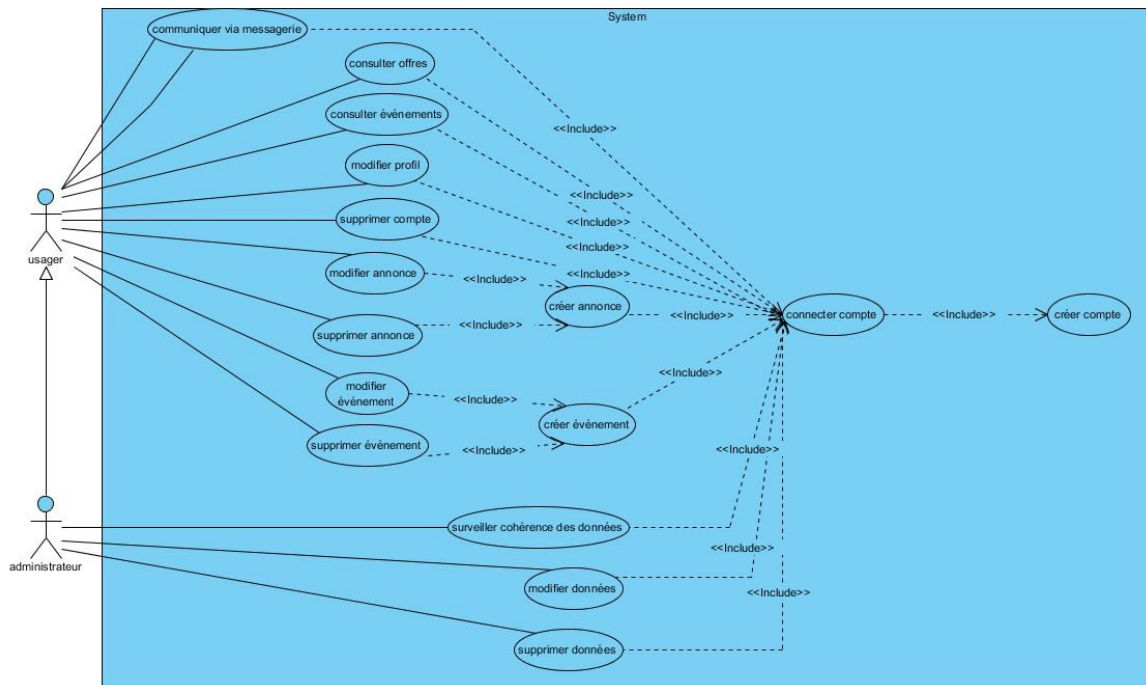


Diagramme Use-Case

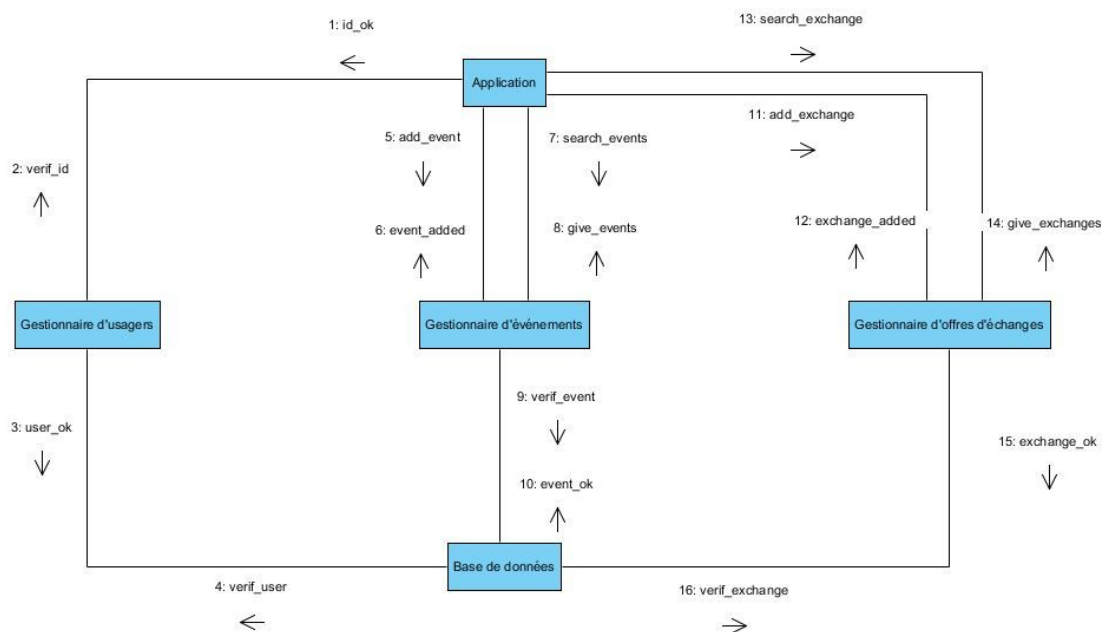
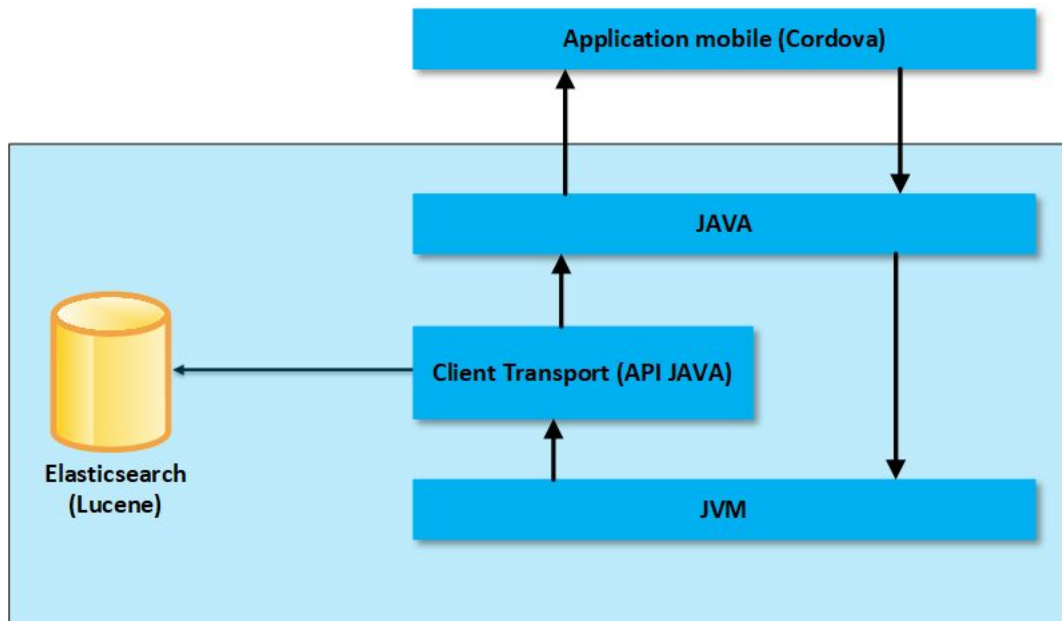
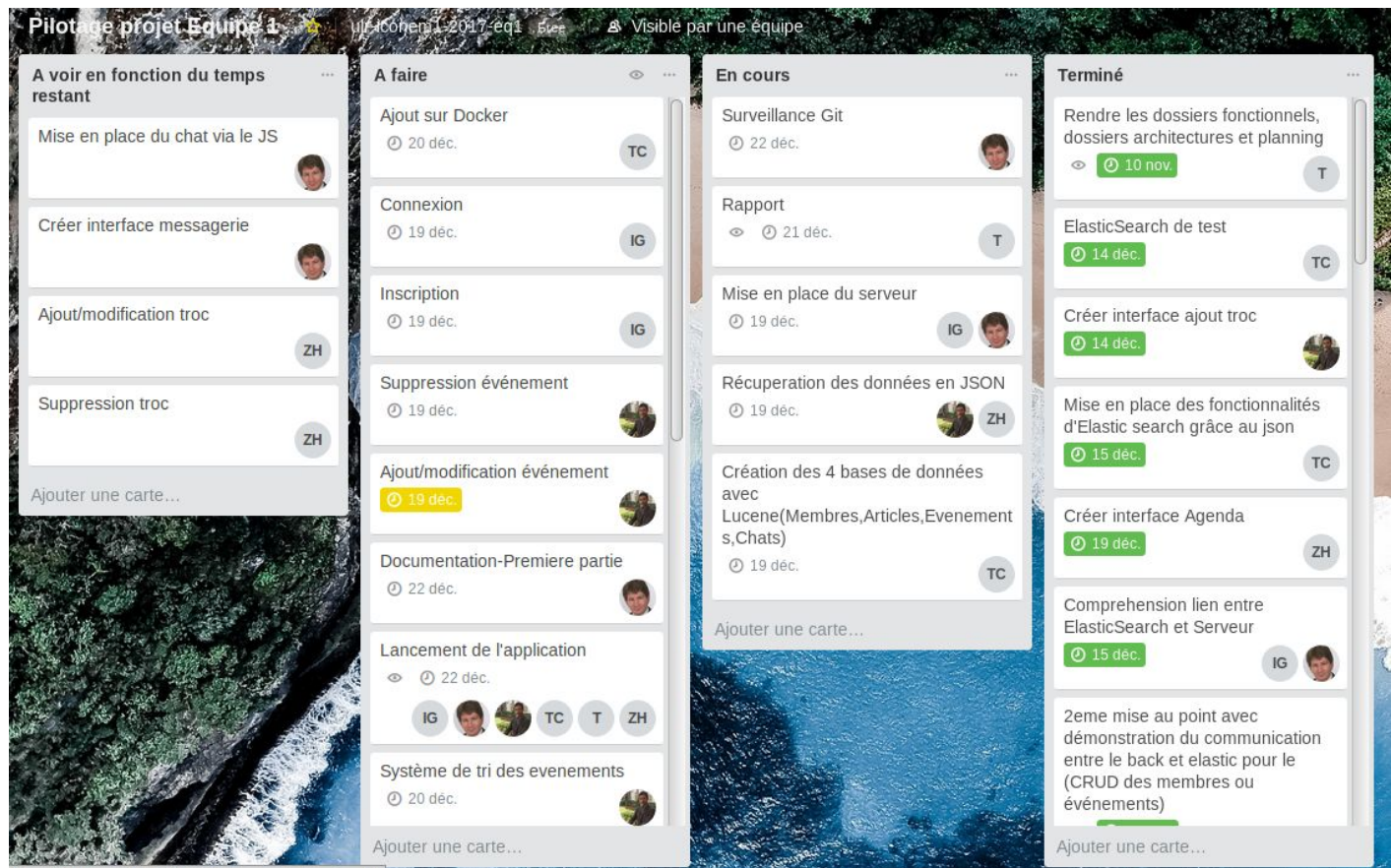


Diagramme de collaboration



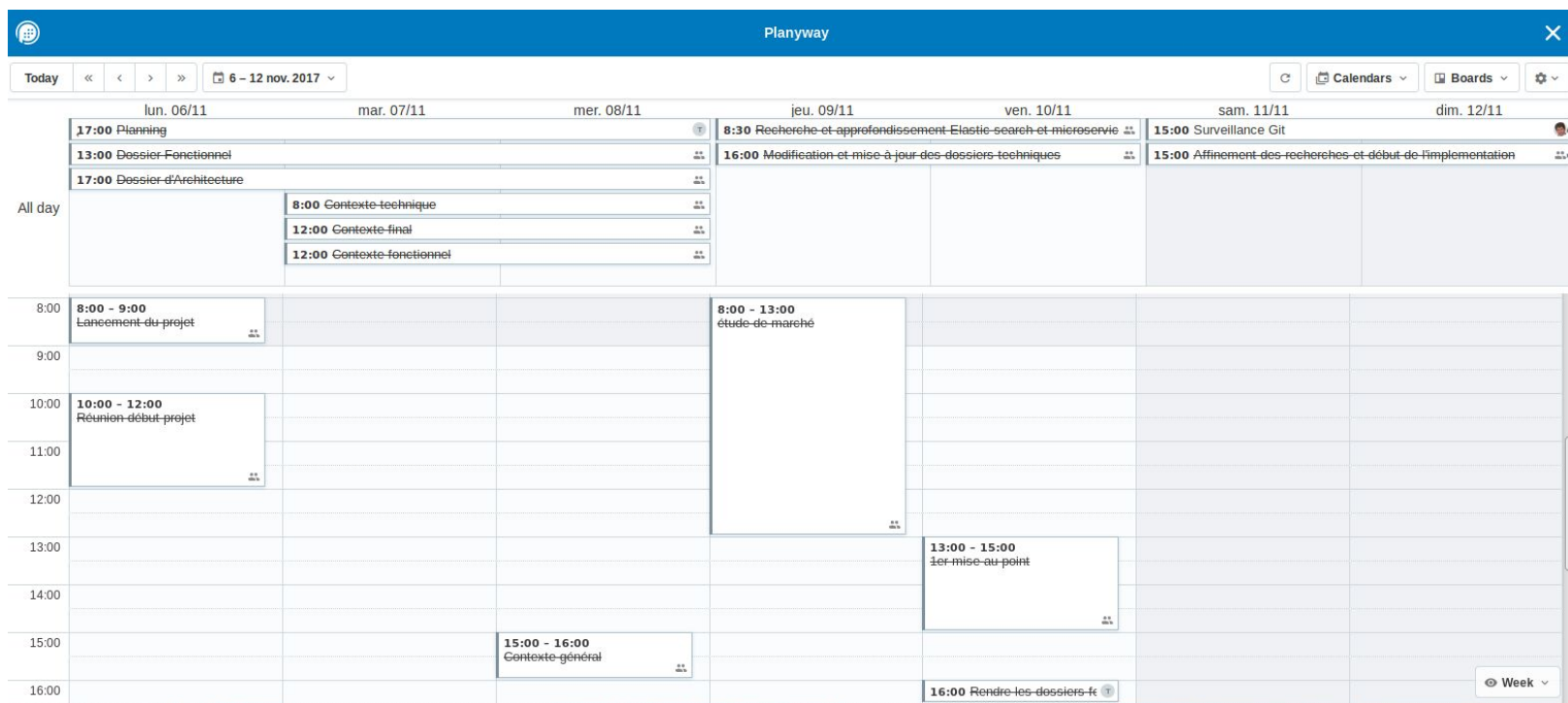
### Architectures logicielles



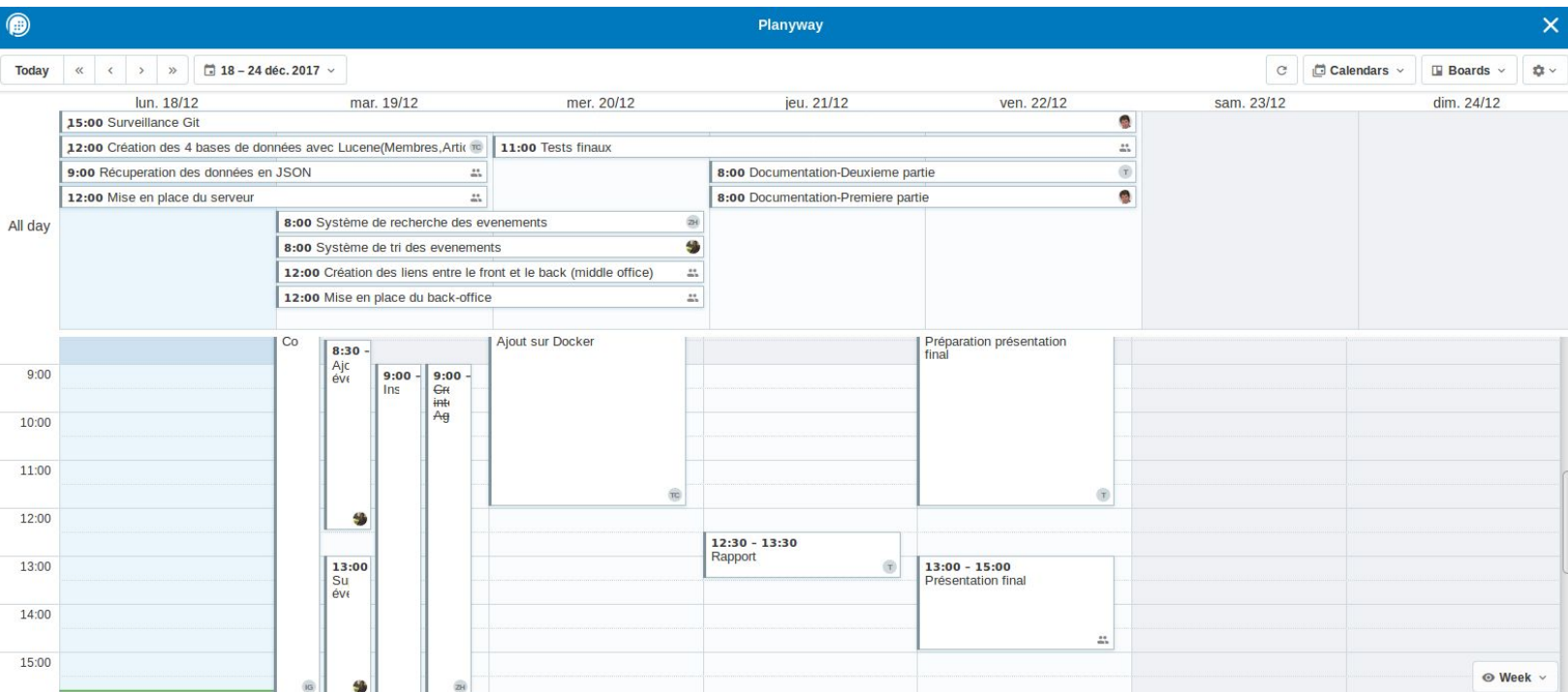
### Trello avec toutes les tâches



# ECO-LEAVES - Rapport



Planning semaine 45



Planning semaine 51

## Recherches / Sources

Nos sources pour installation..etc:

Cordova : <https://cordova.apache.org/docs/en/latest/guide/cli/>

Docker : <https://docs.docker.com/engine/installation/#server>

Elasticsearch : <https://www.elastic.co/guide/en/elasticsearch/client/java-api/current/index.html>

Jaas: <https://docs.oracle.com/javase/8/docs/technotes/guides/security/jaas/JAASRefGuide.html>

Java et l'expression "JEE" : <https://javaee.github.io/javaee-spec/JEE>

Spring : <https://spring.io/>

Nos recherches :

Docker :

<https://lab.engineer.com/blog/Dockerizing-a-simple-website/>

<http://rmpestano.github.io/blog/posts/a-simple-javaee-docker-example.html>

Mise en oeuvre d'une application avec elasticsearch+spring boot+maven:

<http://www.mk Yong.com/spring-boot/spring-boot-spring-data-elasticsearch-example/>

Premier pas avec spring boot et elasticsearch:

<https://dzone.com/articles/first-step-spring-boot-and>

Développement d'une application basée sur une archi microservices:

<http://mbaron.developpez.com/tutoriels/microservices/developpement-application-docker/>

Architecture microservices:

<http://blog.inovia-conseil.fr/?p=155>