Ynov Nantes 04/2019

**Projet DEV**

**Application Viabrico**

Henry MAISONNEUVE Tanguy POTIER

Mathis LE ROY-NIVOT Adrien Vaucard

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



# Introduction

## Contexte

Le projet qui nous a été donné a comme but de concevoir un support de gestion des fournisseurs pour l'entreprise "Viabrico", accessible par ses collaborateurs.

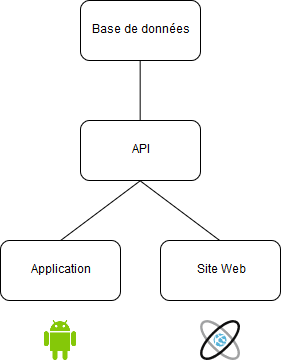
Nous avons donc développé une plateforme qui répond à ces besoins, simple d'utilisation et performante.

## Contraintes

Cette plateforme possède certaines contraintes :

* Elle doit être déclinée sous 2 formes : Un site web et une application
* Une API devra être créée pour accéder à la base de données et pour la connexion
* Le site web doit être développé à l'aide des technologies HTML et JavaScript
* L'application Android doit être faite en Java et consommer l'API via HTTP
* L'API doit être construite en Node JS.

Voici le modèle de notre architecture :



# Choix Techniques

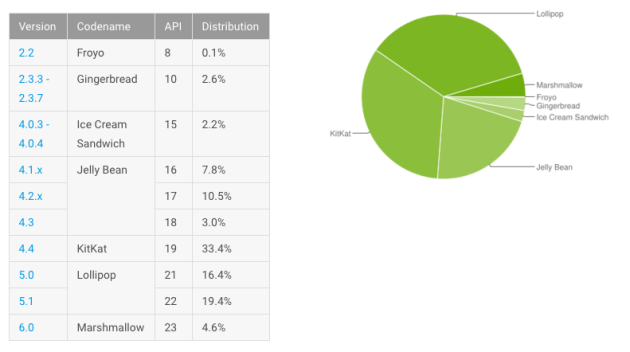
## Application

Pour l'application, la technologie souhaitée était le langage ***Java***.

Android permettant d'utiliser différentes versions d'API, nous avons décider de commencer à partir de l'API 16.

En effet, en regardant la répartition des versions d'Android en Avril 2016, on peut remarquer que 95.1% des appareils utilisent au moins l'API 16 (Jelly Bean). On peut alors imaginer que cette proportion a augmenté durant ces 3 dernières années.

Du fait de nombreux soucis de compatibilité avec des bibliothèques en-deçà de cette version, nous préférons développer avec cette version en API minimum. Cela permettra de cibler un maximum d'appareils, tout en limitant les soucis lors du développement.



## Site Web

Pour le site web, nous avons eu comme contrainte d'utiliser les langages HTML et JS.

Nous avons décidé de ne pas utiliser de frameworks. L'architecture du site n'étant composée que de 3 pages distinctes, il n'était pas judicieux d'utiliser un outil qui organiserait celles-ci.

De plus, utiliser uniquement les technologies Web pures, nous permettait d'avoir une certaine liberté au niveau de méthodes personnalisées, gestion de mémoire, etc…

Ce site web a donc été conçu totalement sur mesure pour répondre au mieux au besoin du client.

## API

L'API que nous avons conçue a été créée avec NodeJS. La documentation pour celle-ci est disponible sur le dépôt Github.

Pour le développement de cette dernière, nous avons décidé de respecter une certaine architecture, dans le but de simplifier la maintenance de celle-ci. De plus, cette architecture s'est vu être utilisée dans de nombreux projets menés par nos biens. Elle est éprouvée est s'est améliorée au fil du temps, pour nous permettre de vous fournir une solution de qualité.

## Base de données

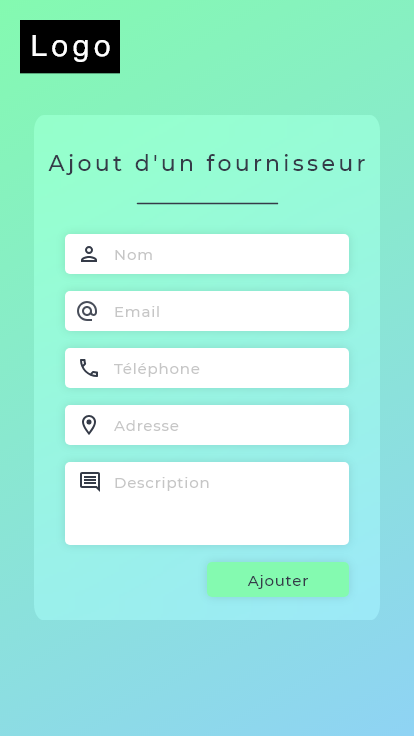
Nous avons décidé d'utiliser le système de gestion de base de données relationnelles MySQL.

Le principal intérêt de ce genre de système est qu'il nous permet de gérer les relations entre les différentes entités de notre base de données.

De plus, l'intérêt d'utiliser MySQL est que ce système de gestion est un des plus performant sur des petits volumes de données. Cela est adapté à la problématique car, dans un premier temps, l'application n'aura à traiter, que les fournisseurs de l'entreprise.

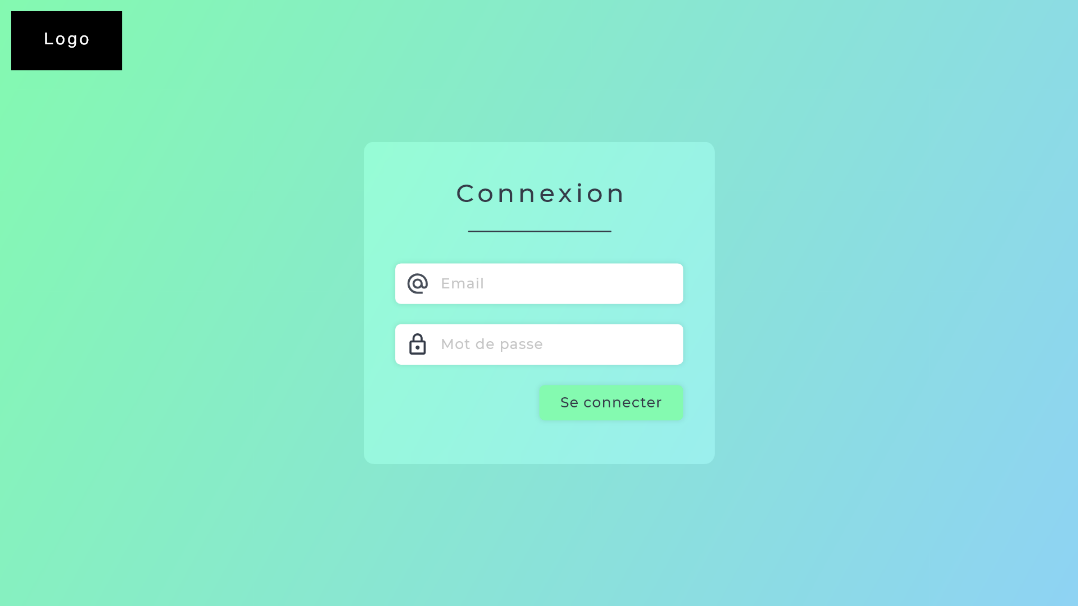
# Solution apportée

## Maquettes

Avant de débuter le développement de cette plateforme, nous avons élaboré des maquettes.

Voici celles que nous avons conçus pour l'application mobile :

Puis, celles du site web :







Ces maquettes nous ont permit d'avoir un modèle sur lequel nous baser pour développer notre plateforme.

## Application

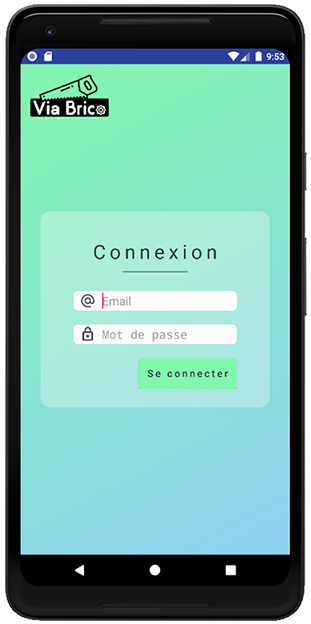
L'application que nous avons développée remplie les fonctions demandées.

Elle nous permet d'ajouter, éditer et supprimer des fournisseurs de l'entreprise concernée.

De plus, un système de connexion sécurisée a été mis en place pour garantir la sécurité des données de l'entreprise.

Ce système permet de se connecter à l'API en utilisant un mot de passe "hashé" (SHA256), ce qui évite de transmettre le mot de passe brute dans les informations circulantes.

Voici quelques visuels de l'application :



## Site Web

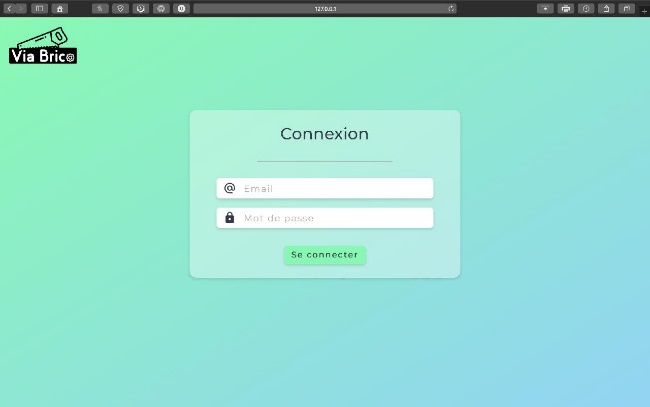
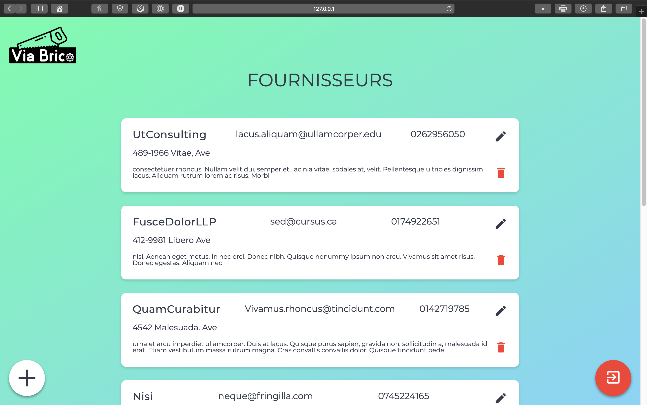
Le site web conçu par notre équipe remplie les mêmes fonctions que l'application. Il dispose aussi du même design que celle-ci (voir maquettes), pour garder une certaine cohérence quant à l'expérience utilisateur.

Ce dernier dispose du même modèle de connexion que l'Application, avec le même niveau de sécurité au niveau des informations qui transitent.

Le système de modification des fournisseurs, quant à lui, diffère de celui de l'application. Cela dans le but de laisser le libre choix à l'entreprise pour le côté visuel. Cela ne change en rien la finalité de l'action.

Le design du site, de même que pour l'application, a évolué par rapport aux maquettes établies au début du projet. Nous avons réfléchi aux améliorations possibles et avons fait le nécessaire pour rendre l'expérience la plus agréable possible.

Voici, des visuels du site Web :





## API

L'API qui nous permet d'accéder à la base de données est conçue en NodeJS et est hébergée sur [Heroku](https://www.heroku.com/) qui est une plateforme en ligne de permettant d'héberger des applications dans le Cloud.

Son principal avantage est qu'elle dispose d'un plan gratuit, ce qui est bénéfique pour tester son application.

A travers celle-ci, nous pouvons récupérer les fournisseurs de l'entreprise, ainsi que se connecter à la plateforme.

C'est le point central de notre plateforme, c'est pourquoi nous avons consacré une certaine partie de notre temps dessus.

## Base de données

Notre base de données, idem que pour l'API, est hébergée sur [Heroku](https://www.heroku.com/) et enregistre les différents fournisseurs de l'entreprise ainsi que les comptes utilisateurs.

Nous accédons à cette dernière via l'API, précédemment décrite.

Dans un respect des lois concernant le RGPD, nous limitons les données enregistrées à propos des utilisateurs. Seuls un identifiant et un mot de passe suffisent pour avoir accès aux fournisseurs.

La création de compte se fera directement depuis la base de données.

# Organisation

## Répartition des tâches

Dans un premier temps, lors d'une réunion "*Brain-Storming*", nous avons délimité les différents axes de travail sur lesquels nous devions travailler. Il en est ressorti 4 :

* La Base de données
* L'API
* L'Application Android
* Le Site Web

Nous nous sommes ensuite réparti les différentes tâches pour que l'on puisse avancer séparément, chacun de notre côté.

L'API avec la base de données ont été une priorité, car cela était le cœur de notre application et sans elles, nous ne pouvions continuer à avancer. Nous avons donc tous contribuer à l'établissement et au bon fonctionnement de ces 2 aspects techniques.

Ensuite, nous avons établi les maquettes de notre Application, ainsi que celles pour le Site Web. Cela nous a permis d'avoir une base sur laquelle travailler et une idée du rendu final, ce qui nous a bien orienté pour la suite du projet.

S'en est suivi par la suite le côté développement. En se répartissant par équipe de 2, nous gardions une équité et divisions la tâche en affectant 2 personnes sur l'Application et 2 autres sur le site Web.

Une fois nos rôles terminés, nous avons mis en commun notre travail et proposé des améliorations. Nous avons, chacun, veillé au fonctionnement, performances et sécurité des outils développés.

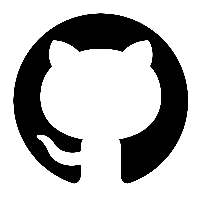
Nous livrons donc la V 1.0 de notre plateforme, qui dispose de toutes les fonctionnalités demandées dans le cahier des charges. Celle-ci a été vérifiée et testée et est prête pour un déploiement.

Nous avons préféré nous focaliser sur la stabilité et la sécurité de notre plateforme plutôt que sur des fonctionnalités en plus car nous favorisons un produit condensé fini plutôt qu'une plateforme complète mais instable.

Notre architecture de projet ayant été réfléchie auparavant pour la maintenance et l'évolutivité du produit, il sera aisé d'ajouter des fonctionnalités à cette solution. Nous avons d'ailleurs commencé à réfléchir à l'implémentation de nouvelles fonctions.

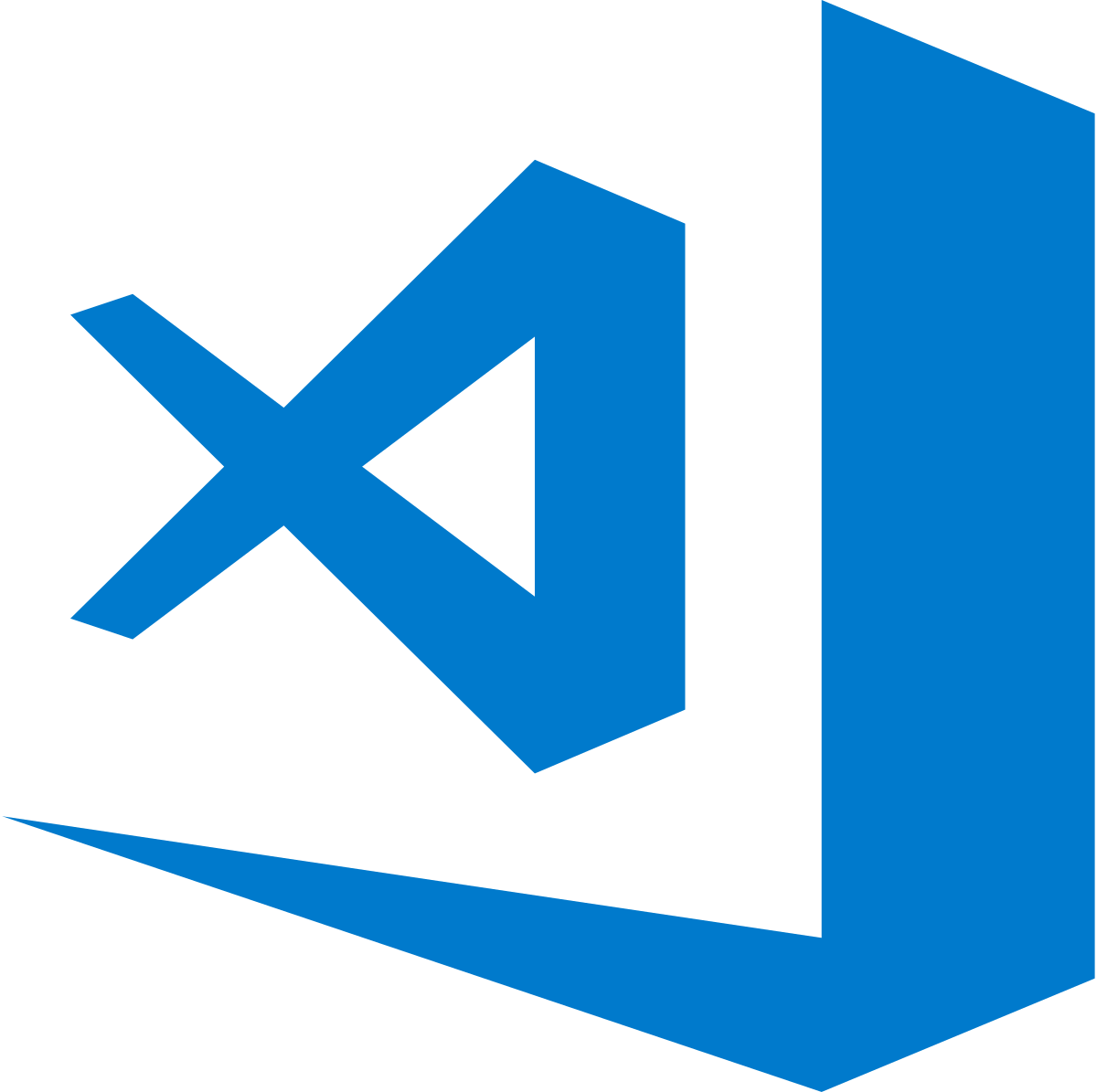
Si vous souhaitez continuer le projet à nos côtés, nous pourrions livrer une 2ème version, plus développée et complète, permettant de faciliter encore plus la gestion de vos collaborateurs.

## Outils utilisés



Un dépôt Github nous permettait de centraliser tout notre travail et d'avoir une certaine transparence quant à l'avancée de chacun.

Il nous permet aussi d'avoir un visuel sur l'historique du projet, pour ainsi voir les erreurs corrigées et les pistes écartées.



L'ensemble de l'équipe de projet travaillait avec l'IDE *Visual Studio Code*, qui est performant, stable et dispose d'une multitude d'extensions pour améliorer le développement.

# Annexes

## Liens

API : <https://viabrico-api.herokuapp.com/>