POC SUPINFO

*Sommaire*

[Description 2](#_Toc61461500)

[Solutions techniques 3](#_Toc61461501)

[Choix de la base de donnée NoSQL 4](#_Toc61461502)

[MongoDB 4](#_Toc61461503)

[Choix des outils de développement 5](#_Toc61461504)

[Angular 5](#_Toc61461505)

[NodeJS 5](#_Toc61461506)

[Express 6](#_Toc61461507)

[Mongoose 6](#_Toc61461508)

[Représentation de l’architecture 6](#_Toc61461509)

[Annexe 7](#_Toc61461510)

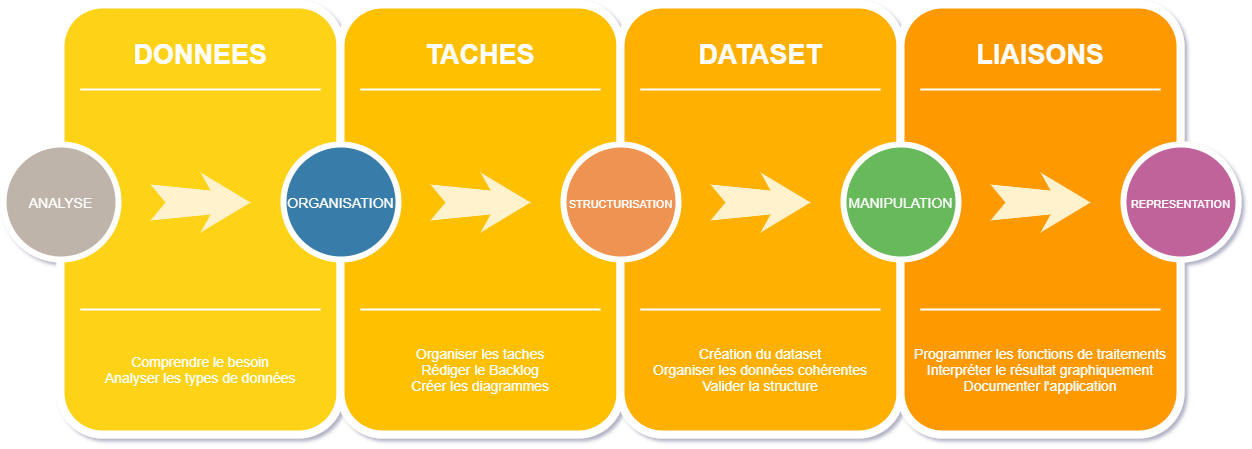
# Description

Notre projet consiste à répondre à un besoin crucial de la part de nos clients. Actuellement en pleine étude de marché, ils aimeraient pouvoir représenter, sous forme visuel, des données concernant tous les étudiants de l’école Supinfo. Ils aimeraient par exemple connaître le taux de réussite en fonctions des Campus, le nombre de recrutement de la part des entreprises externes, le pourcentage de présence dans chaque Campus…

Ceux à quoi nous avons répondu, nôtre tache s’applique sur la construction des données relatives aux étudiants et aux écoles et nos actions portent sur la représentation des besoins de nos clients.

Pour cela nous allons utiliser plusieurs outils techniques et langage de programmation que nous allons vous présenter au fil de ce PoC.

# Solutions techniques

La représentation de données s’effectue la plupart du temps sous formes de graphiques, peu importe le type, sur lequel les données sont transformées assez souvent en pourcentage ou numériquement. Sur ces conditions nous avons actionné une analyse de traitement efficace.

Après avoir effectué ce traitement nous avons fourni une maitrise technique qui se base sur la représentation.

Pour les représenter nous avons fait appel à des fonctions disponibles dans le langage R, un coté serveur pour héberger notre application et un coté front pour représenter nos requêtes.

Notre application est disponible sur ce repo ici présent : <https://github.com/Yokai1000/5Data>

Nous allons voir à quoi correspond chaque technologie que nous avons utilisées durant le projet.

# Choix de la base de données NoSQL



## MongoDB

MongoDB est sous licence SSPL et est développé par MongoDB Inc. C’est un outil de gestion de base de données NoSQL (Not only SQL) orienté document.

La base de données MongoDB est composée de collections contenant des documents de type BSON (JSON en binaire). Elle permet aux documents d’avoir différents champs et structures et permet de repartir les données en un ensemble de paires clé-valeur.

Nous avons choisi MongoDB pour plusieurs raisons :

* Grâce à sa flexibilité ; il ne nécessite pas de schéma prédéfini. Vous pouvez modifier les schémas dynamiquement.
* Il nous a paru efficace grâce à son interface intuitive.
* Les documents sont créés sans les contraintes de relation et sont totalement indépendants.
* Il propose un large panel d’importation de données
* Il dispose d’une infrastructure cloud qui vous permettra par la suite de vous digitaliser facilement
* Il est gratuit (open-source)
* Il permet un accès simple aux données stockées, saisies et la récupération
* Il peut s’exécuter sur plusieurs serveurs et ainsi éviter les cas de pannes
* Il peut prendre en charge MapReduce et des outils d’agrégation flexibles.

# Choix des outils de développement

Une image contenant texte, clipart

Description générée automatiquement

## Angular

Angular est un framework d’application web frontale écrit en JavaScript par Google.

Il permet de créer et développer des applications web dynamique qui communiquent avec un serveur principal. Il dispose de fonctionnalités et d’outils qui simplifient le développement des applications et garantie d’excellents résultats de performance.

Nous l’avons choisi pour sa facilité d’installation et son efficacité de rendu en composant.

Il nous a permis de décomposer notre application en composant et de pouvoir affecter à celle-ci des rendues différents en fonction de leur utilité.

Il a également l’avantage de comprendre le JSON qu’on retrouve dans MongoDB.

Une image contenant texte, clipart

Description générée automatiquement

## NodeJS

NodeJs est une plateforme de développement orienté serveur. Il permet d’exécuter du Javascript dans un environnement à faible latence et à débit élevé, tout en adoptant une approche ‘non bloquante’ pour traiter les demandes.

Nous l’avons choisi pour plusieurs critères :

* Il conserve le même langage et le même format de données qu’Angular
* Il permet à notre application d’être héberger en local
* Il permet d’utiliser facilement les bases de données NoSql comme MongoDB
* Il permet de gérer plusieurs requêtes en même temps
* Il compile avant d’exécuter, ce qui permet d’avoir de bonnes performances
* Il permet d’intégrer un serveur http

API ??



## Express

ExpressJs est un framework qui permet d’avoir un cadre minimal et flexible de NodeJs et d’avoir des fonctionnalités robustes pour l’utilisation des applications.

Nous l’avons choisi car :

* Il nous aide à créer des applications Web plus rapides et intelligentes
* Il hérite des fonctionnalités de NodeJs : flexible, évolutif et bonnes performances
* Il contient un mécanisme de routage très avancé
* Il facilite le débogage en identifiant la partie qui contient des erreurs



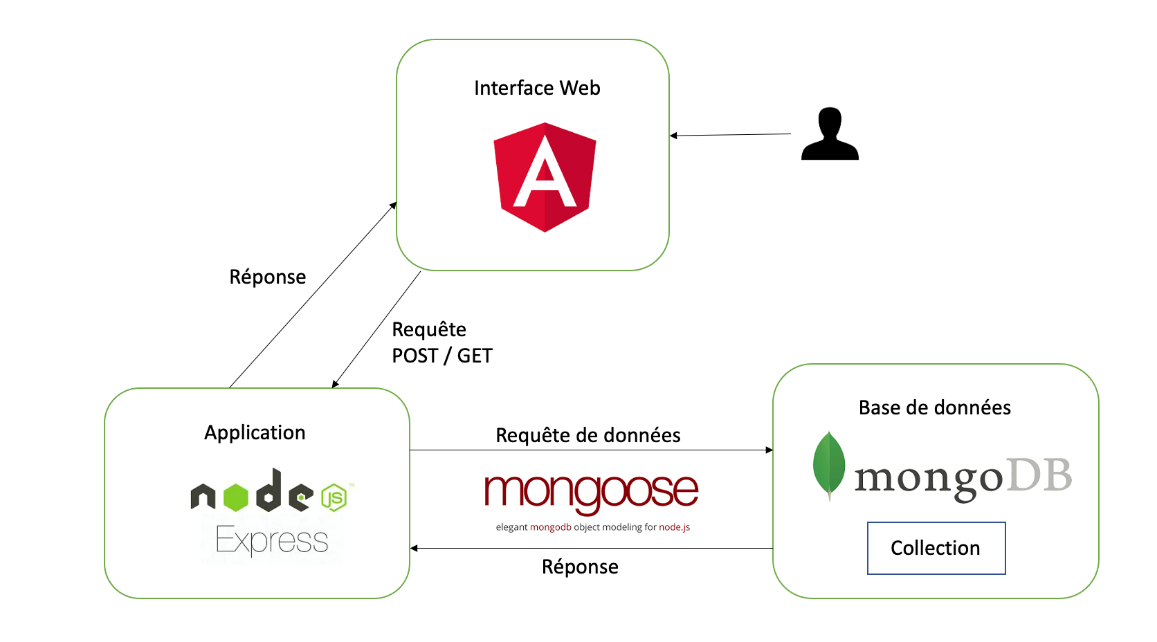
## Mongoose

Mongoose est une librairie MongoDB ODM (Object Data Modeling) basé sur Node.Js.

MongoDB étant flexible, Mongoose permet d’appliquer un schéma spécifique au niveau de l’application pour faciliter le travail.

Une fois que nous avons défini un modèle Mongoose, nous pouvons exécuter des requêtes pour récupérer, mettre à jour ou supprimer des données dans une collection.

# Représentation de l’architecture

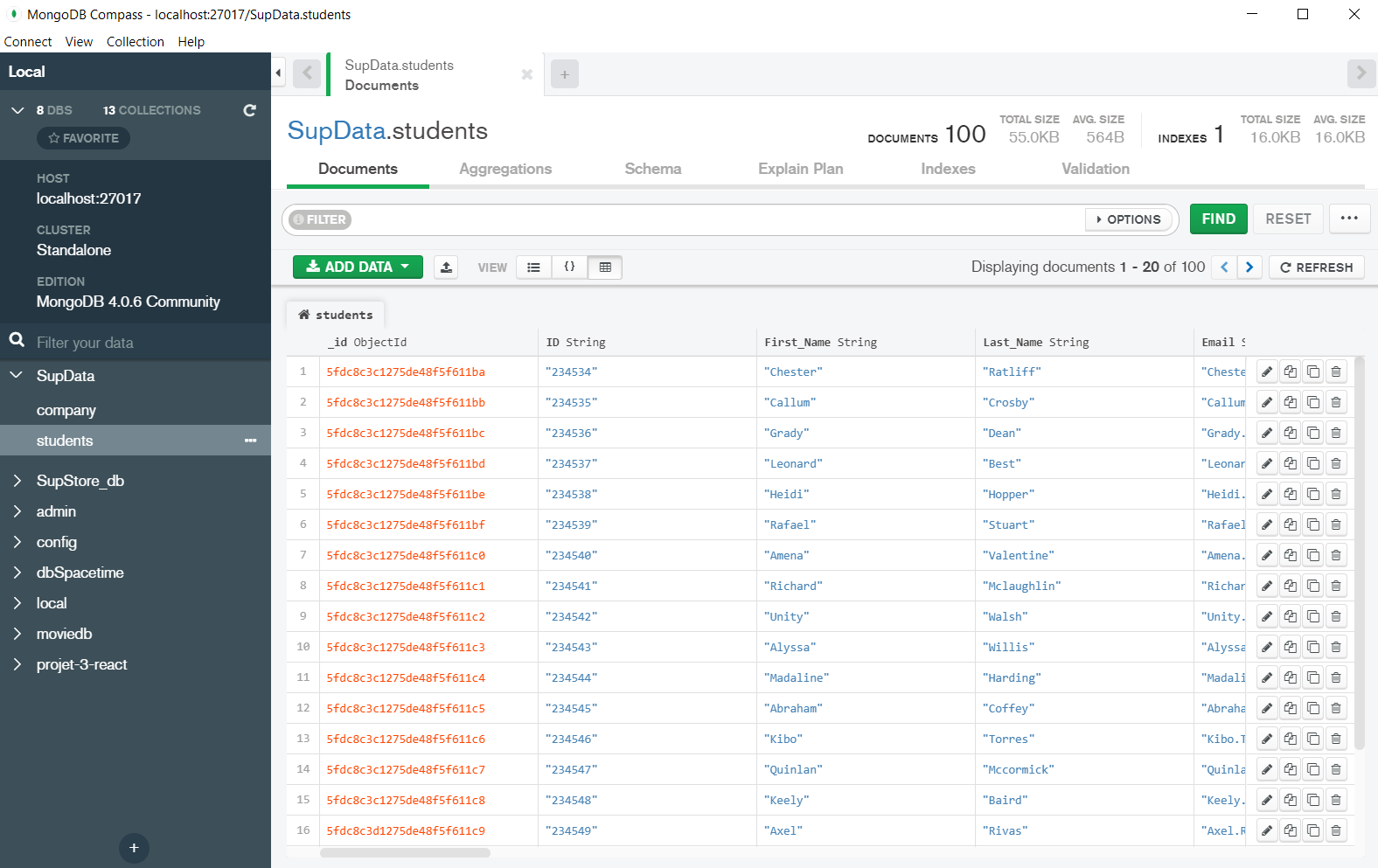


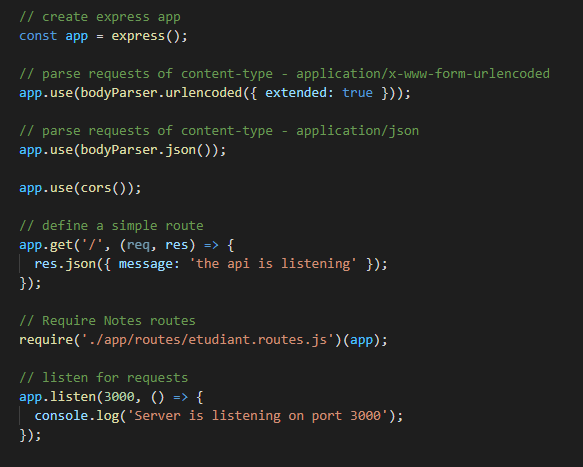
Grace à tout ces outils nous avons pu développer un API REST. C’est-à-dire, un site web qui manipule une base de données et qui nous renvoi des données relatives à une requête spécifique.

# Annexe

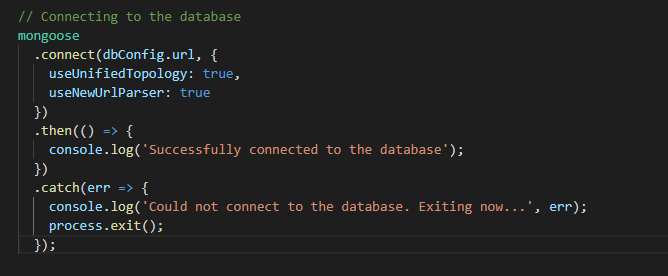
Une image contenant texte

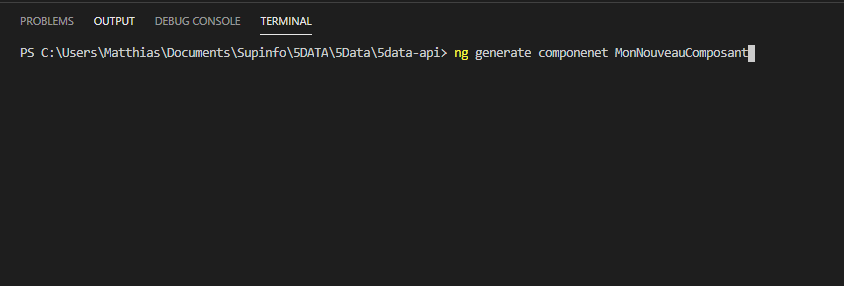
Description générée automatiquementCapture NodeJS

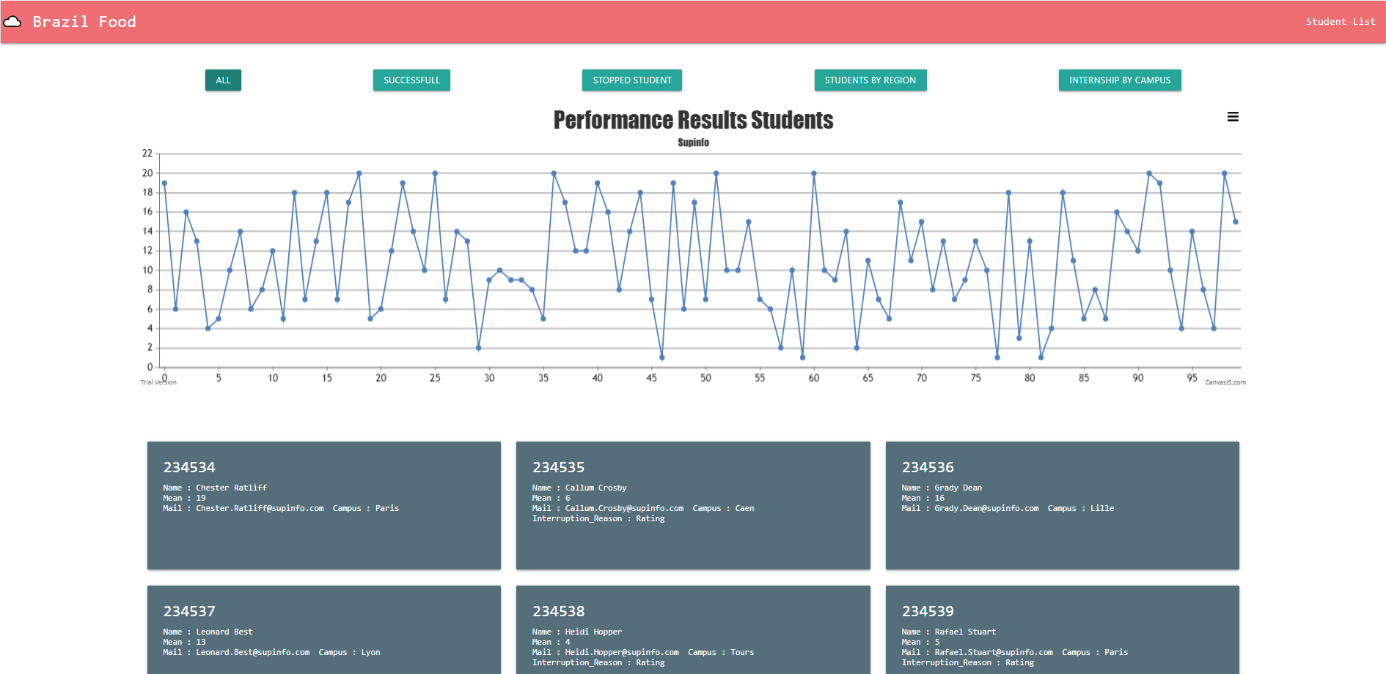
Interface MongoDB



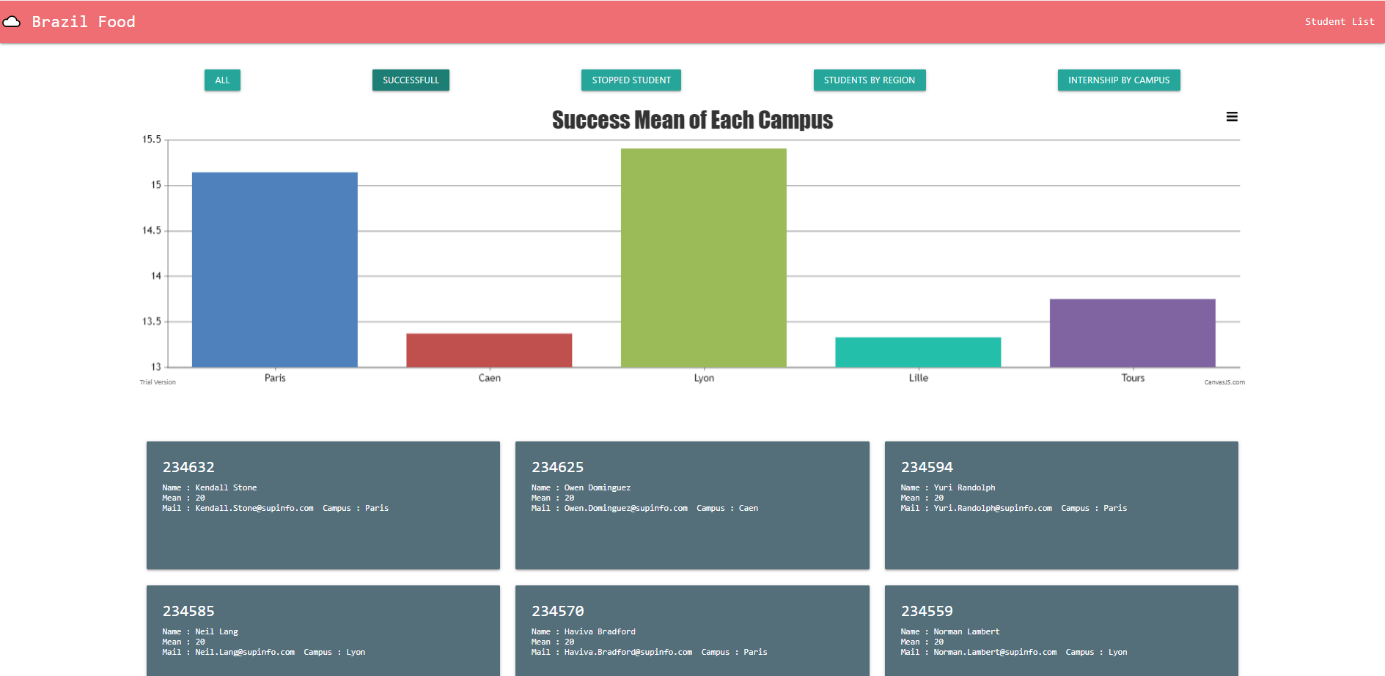
Express Initialisation

Mongoose Initialisation

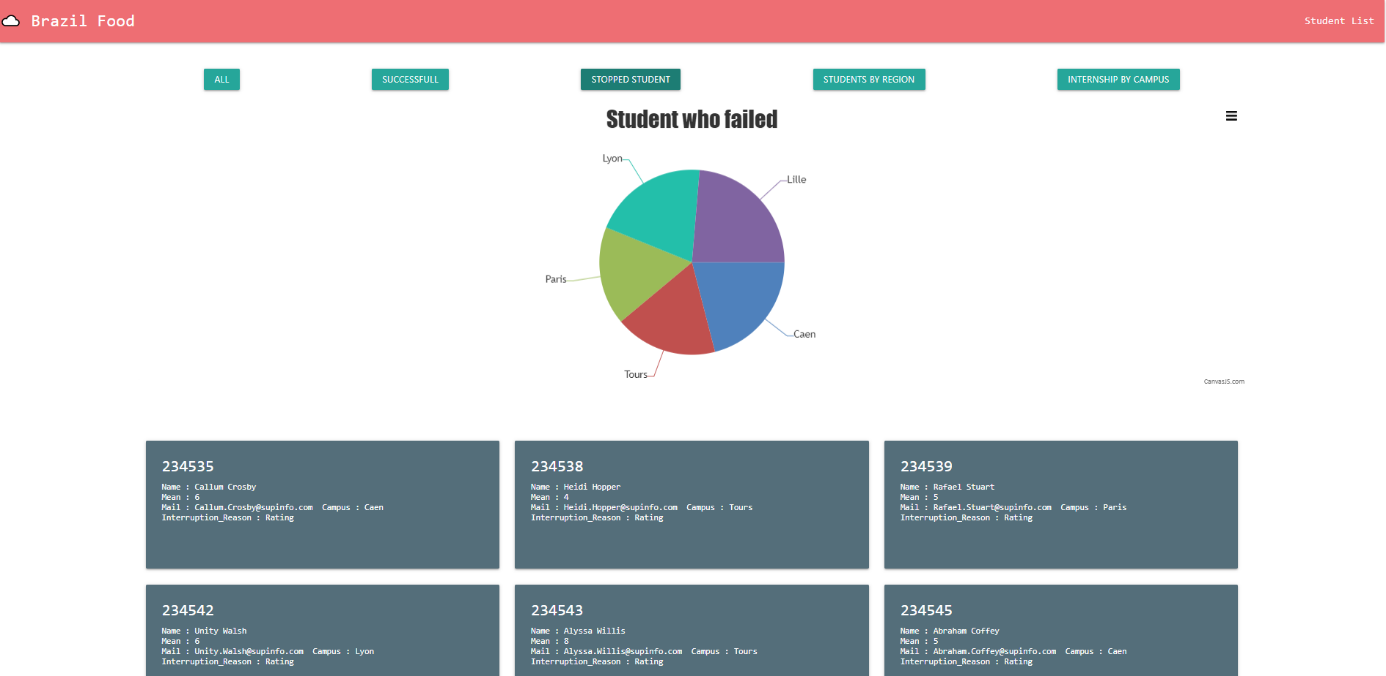
Création d’un nouveau composant Angular



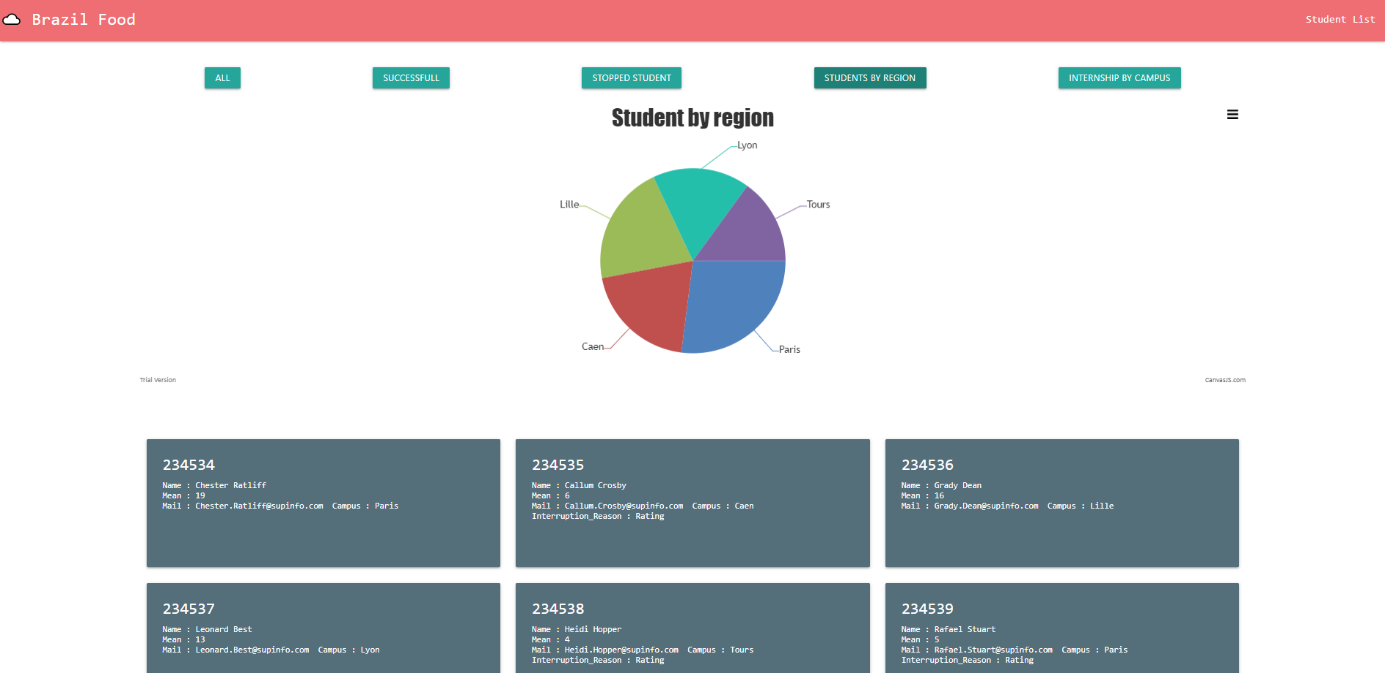
Représentation de tous les étudiants sur l’application



Représentation des meilleurs étudiants par campus sur l’application



Représentation des étudiants en échec par campus sur l’application



Représentation des étudiants en échec par campus sur l’application

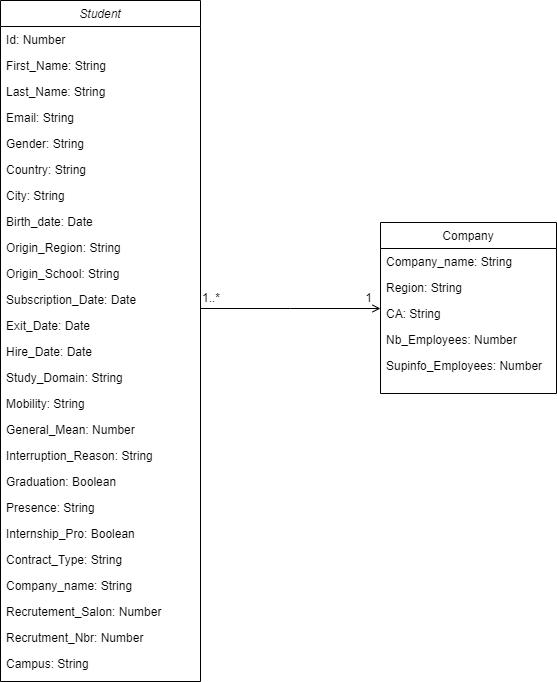


Diagramme de classe Student & Company