

Projet M2 MIAGE

EVNTS



CARON Baptiste  
AUBERT Vincent  
CAMACHO Romain  
GITTEAU Sidney

2023/2024

I.	Présentation du sujet.....	3
II.	Architecture de votre application.....	5
1.	Organisation de l'application .....	5
2.	Diagrammes BDD .....	6
a.	BDD Service Google .....	6
b.	BDD Contact.....	6
c.	BDD Gestion Salon .....	7
d.	BDD Gestion Utilisateur .....	7
e.	BDD Authentification .....	8
III.	Fonctionnalités opérationnelles .....	8
1.	Service Authentification.....	9
2.	Service Salon .....	10
3.	Service GestionUtilisateur.....	11
4.	Service Contact .....	11
5.	Service Google.....	11
IV.	Guide utilisateur .....	14

## I. Présentation du sujet

Le projet EVNTS est une application permettant l'organisation d'évènements par le biais de salons. De la même manière que l'application Discord, EVNTS propose la création de salon personnalisés, appartenant à une personne. Dans ce salon, celui-ci peut désigner des modérateurs qui auront des autorisations supplémentaires contrairement à un utilisateur du salon normal. Le propriétaire du salon ainsi que les modérateurs qu'il a choisi pourront créer des évènements à l'intérieur de ce salon.

Par exemple je souhaite créer un salon destiné à l'organisation de concours d'échec. Je vais pouvoir créer un Salon "Echec Master Assembly" (par exemple). Je vais pouvoir inviter des collègues à rejoindre ce salon.

Une fois dans ce salon je pourrais communiquer à l'aide de la messagerie contenue dans mon salon.

Le salon a été créé. Je peux maintenant créer des évènements à l'intérieur de mon salon, avec une date, un lieu, un descriptif et le nombre requise de personnes pour participer à cet Event.

J'ai donc créé un évènement "Chess Tour 2024". Les personnes du salon pourront rejoindre mon évènement et discuter à l'intérieur de la messagerie de cet évènement.

Une fois le nombre de personne requises d'inscrites pour l'évènement, il doit être validé par le créateur du salon ou un de ces modérateurs.

Une fois l'évènement terminé, celui-ci est supprimé.

L'application utilise consul et est entièrement dockerisée.

## Services 6 total

Search Across

Health Status

Service Type

consul

1 Instance

authentication

1 Instance

gateway

1 Instance

gestionGoogle
































































1 Instance

gestionSalon

1 Instance

gestionUtilisateur

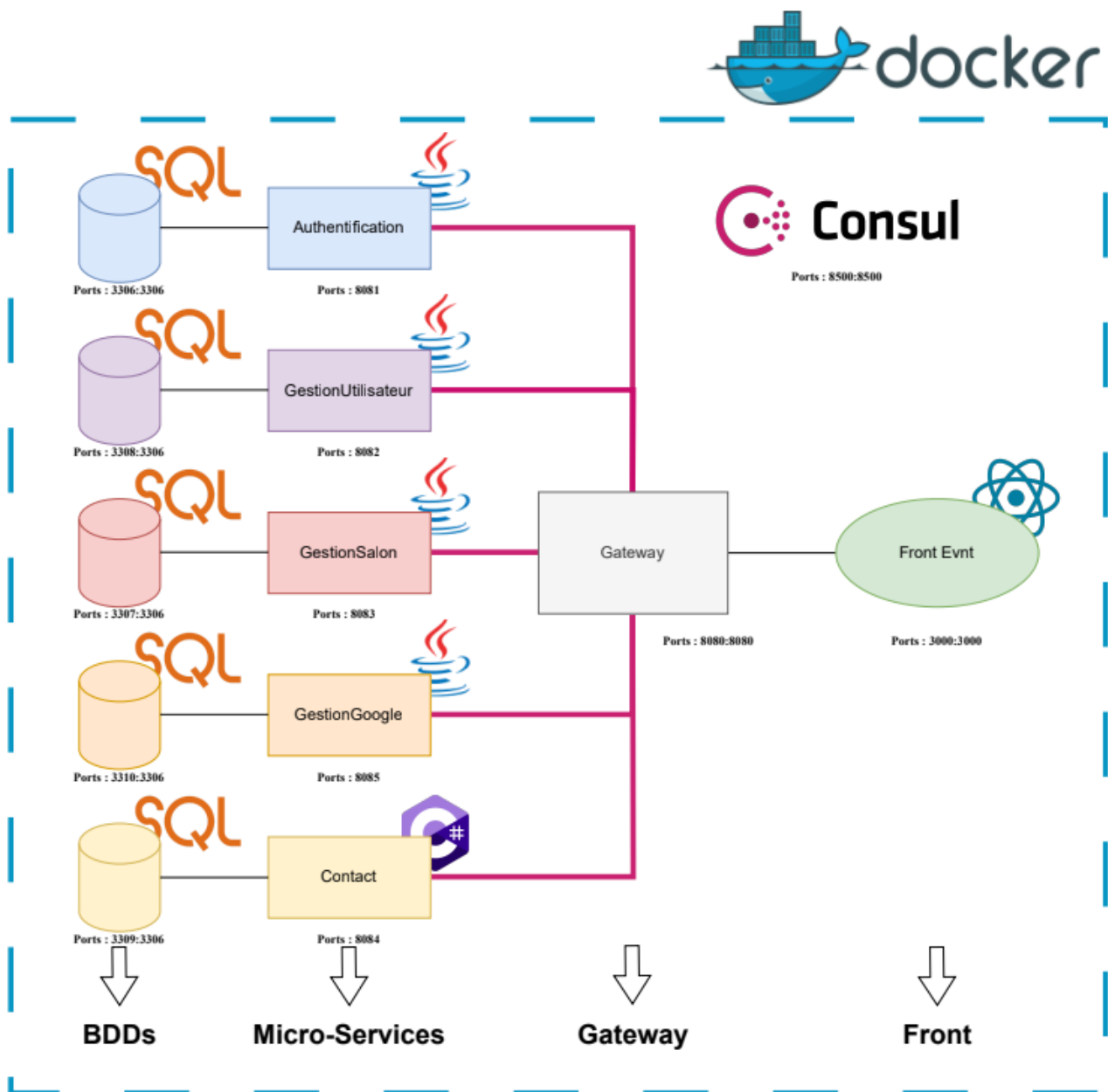
1 Instance

<input type="checkbox"/>	▼  evnts		Running (12/1)	12.5%	2 seconds ago		:	
<input type="checkbox"/>	 <a href="#">dbGestionUtilisateur</a> 69143b2c2567 	<a href="#">mysql:latest</a>	Running	0.28% <a href="#">3308:3306</a> 	2 minutes ago		:	
<input type="checkbox"/>	 <a href="#">frontend-1</a> c019ef36a758 	<a href="#">evnts-frontend</a>	Running	9.27% <a href="#">3000:3000</a> 	2 minutes ago		:	
<input type="checkbox"/>	 <a href="#">consul-1</a> 20663664552e 	<a href="#">consul:1.9</a>	Running	0.87% <a href="#">8500:8500</a> 	2 minutes ago		:	
<input type="checkbox"/>	 <a href="#">dbContact-1</a> 80ea8382a42f 	<a href="#">mysql:latest</a>	Running	0.25% <a href="#">3309:3306</a> 	2 minutes ago		:	
<input type="checkbox"/>	 <a href="#">dbSalonName</a> 8d5be155e0db 	<a href="#">mysql:latest</a>	Running	0.25% <a href="#">3307:3306</a> 	2 minutes ago		:	
<input type="checkbox"/>	 <a href="#">dbGoogle</a> fb96a36eeef94 	<a href="#">mysql:latest</a>	Running	0.26% <a href="#">3310:3306</a> 	2 minutes ago		:	
<input type="checkbox"/>	 <a href="#">googleApp-1</a> 3360d3dc5c87 	<a href="#">mon-application-google</a>	Running	0.29%	2 minutes ago		:	
<input type="checkbox"/>	 <a href="#">salonApp-1</a> c480ace9a933 	<a href="#">mon-application-salon</a>	Running	0.18%	2 minutes ago		:	
<input type="checkbox"/>	 <a href="#">authenticationApp-1</a> 5a4abd6c0adc 	<a href="#">mon-application-authentication</a>	Running	0.25%	2 minutes ago		:	
<input type="checkbox"/>	 <a href="#">gateway-1</a> df9e6867a65b 	<a href="#">mon-application-gateway</a>	Running	0.14% <a href="#">8080:8080</a> 	2 minutes ago		:	
<input type="checkbox"/>	 <a href="#">gestionUtilisateurApp-1</a> b14366706d79 	<a href="#">mon-application-utilisateur</a>	Running	0.4%	2 minutes ago		:	
<input type="checkbox"/>	 <a href="#">contactApp-1</a> 7f0bf6231113 	<a href="#">mon-application-contact</a>	Restarting (13	0%	28 seconds ago		:	
<input type="checkbox"/>	 <a href="#">dbAuthentication</a> d9cac63e28bb2 	<a href="#">mysql:latest</a>	Running	0.51% <a href="#">3306:3306</a> 	2 minutes ago		:	

Une erreur remonte dans le container de l'application .NET

## II. Architecture de votre application

### 1. Organisation de l'application



Ce projet est constitué de 5 micro-services, 5 BDD SQL, une gateway et 1 front. L'application est entièrement dockerisée et contient consul.

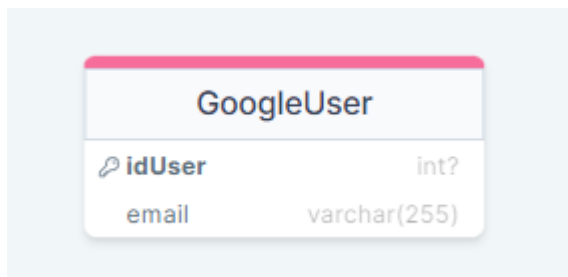
Les microservices java sont des applications Spring Boot, mappées sur les ports 8081,8082,8083,8085.

Le microservice Contact est en .NET mappé sur le port 8084. Chaque application possède sa BDD SQL.

Le consul est sur le port 8500, la gateway est sur le port 8080 et gère la redirection des services. L'application front end est en REACT.

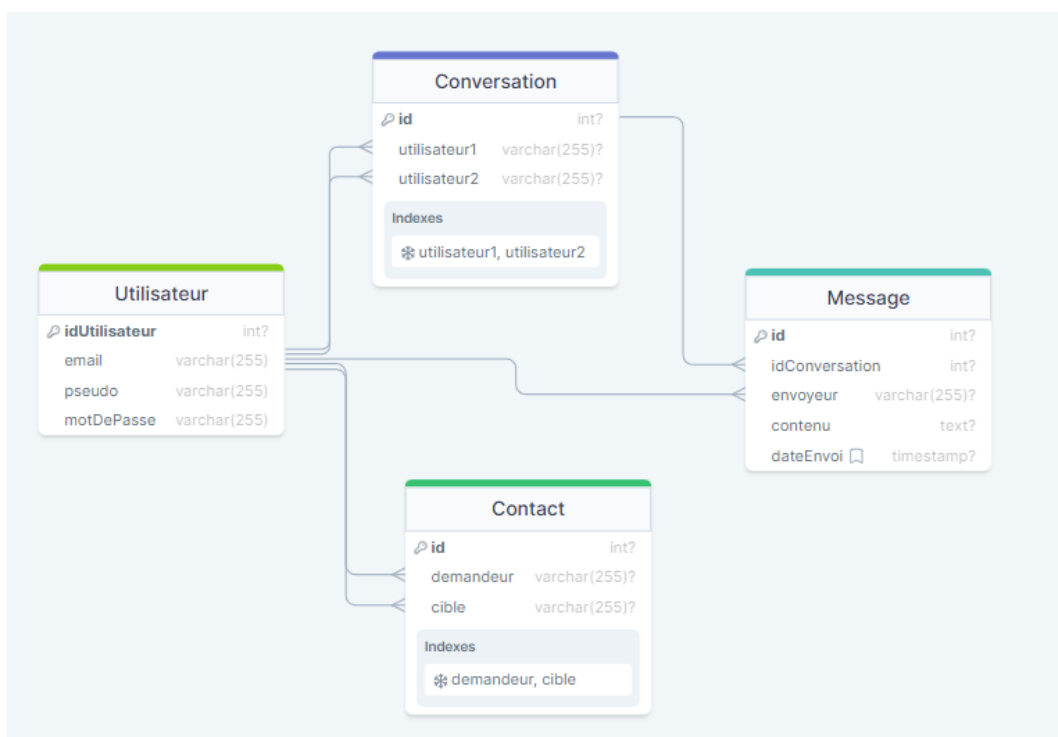
## 2. Diagrammes BDD

### a. BDD Service Google



Une BDD simple avec une table et deux champs afin de récupérer l'email de l'utilisateur.

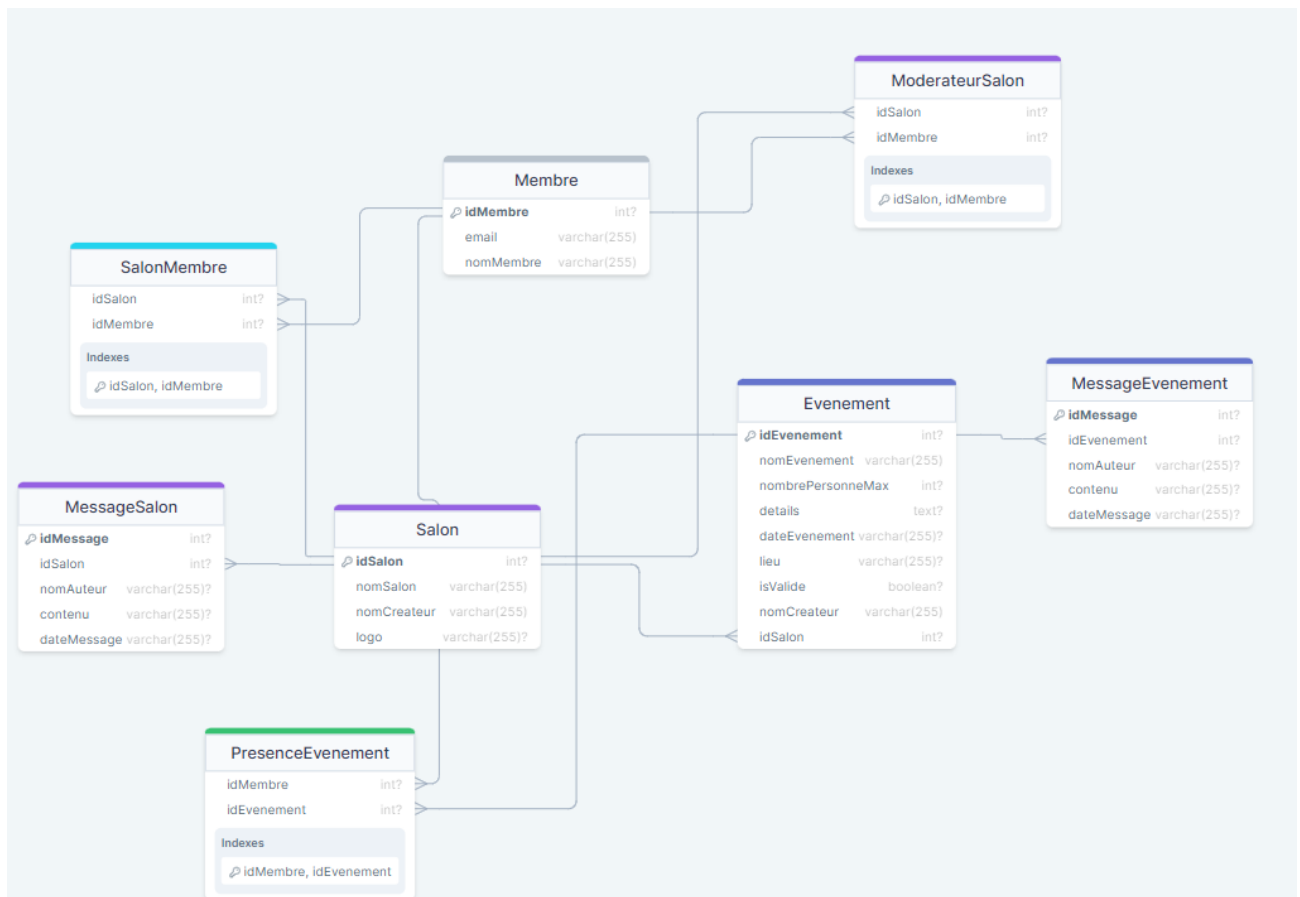
### b. BDD Contact



Les tables définies sont conçues pour stocker les données d'un système de messagerie ou de communication entre utilisateurs. La table Utilisateur enregistre les informations de base sur les utilisateurs, identifiés par leur adresse e-mail. La table Contact gère les relations entre utilisateurs, tandis que la table Conversation stocke les détails des conversations entre

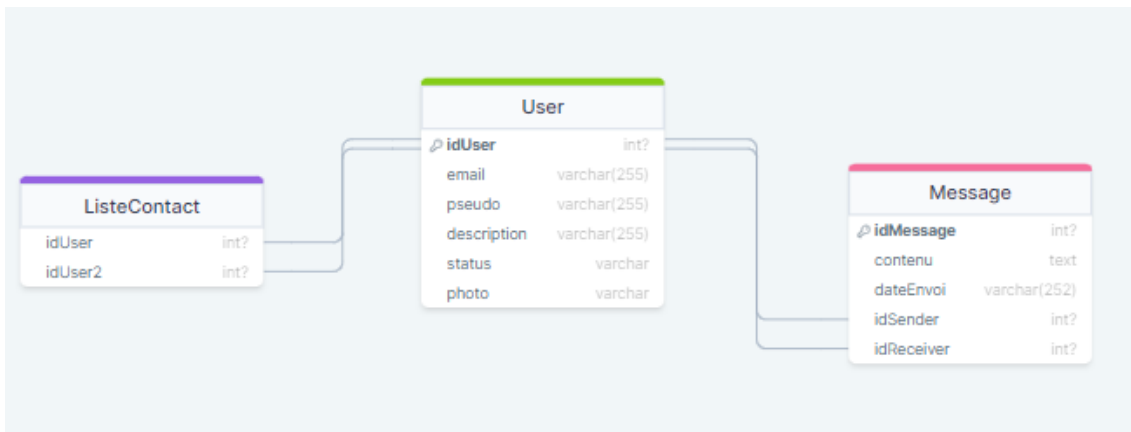
eux. Enfin, la table Message enregistre les messages échangés, associés à une conversation spécifique. Des contraintes de clé étrangère sont utilisées pour garantir l'intégrité des données, assurant que les adresses e-mail et les conversations référencées existent bien dans les tables associées. Cette structure de base permet de gérer efficacement les interactions et les échanges entre les utilisateurs du système.

### c. BDD Gestion Salon



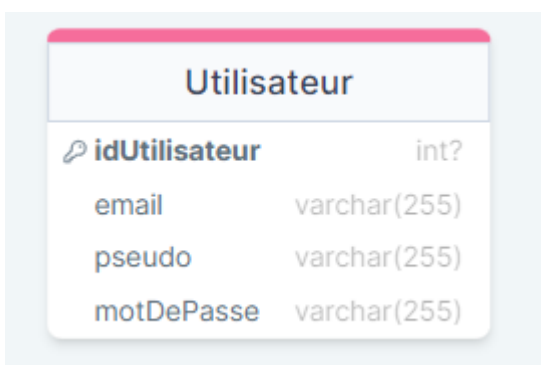
Ces tables sont conçues pour gérer un système de salons virtuels et d'événements associés. La table Membre conserve les informations sur les utilisateurs, tandis que la table Salon stocke les détails des salons, tels que leur nom, leur créateur et leur logo. Les tables SalonMembre et ModerateurSalon sont des tables de liaison pour associer les membres aux salons et désigner des modérateurs respectivement. La table Evenement enregistre les événements planifiés, avec des détails comme le nom, la date, le lieu et le nombre maximal de participants. Les tables PresenceEvenement, MessageSalon et MessageEvenement gèrent respectivement la présence des membres aux événements, les messages échangés dans les salons et ceux échangés spécifiquement lors d'événements. Cette structure de base permet de gérer efficacement les interactions et les événements au sein du système de salons virtuels.

### d. BDD Gestion Utilisateur



La table User est conçue pour stocker les informations relatives aux utilisateurs d'un système. Chaque utilisateur est identifié par un ID unique, et des détails tels que leur adresse e-mail, pseudo et description sont enregistrés. La description est définie comme une valeur obligatoire pour chaque utilisateur. Le statut et la photo de l'utilisateur sont également pris en charge, bien qu'ils ne soient pas obligatoires. Cette structure de base permet de gérer efficacement les profils des utilisateurs dans le système.

#### *e. BDD Authentication*



La table Utilisateur est créée dans la base de données "authentification" pour gérer les informations des utilisateurs. Chaque utilisateur est représenté par un ID unique (idUtilisateur) généré automatiquement à l'aide de l'auto-incrémentation. Les champs email, pseudo et motDePasse sont définis comme non nuls pour garantir que chaque utilisateur a des informations d'identification valides.

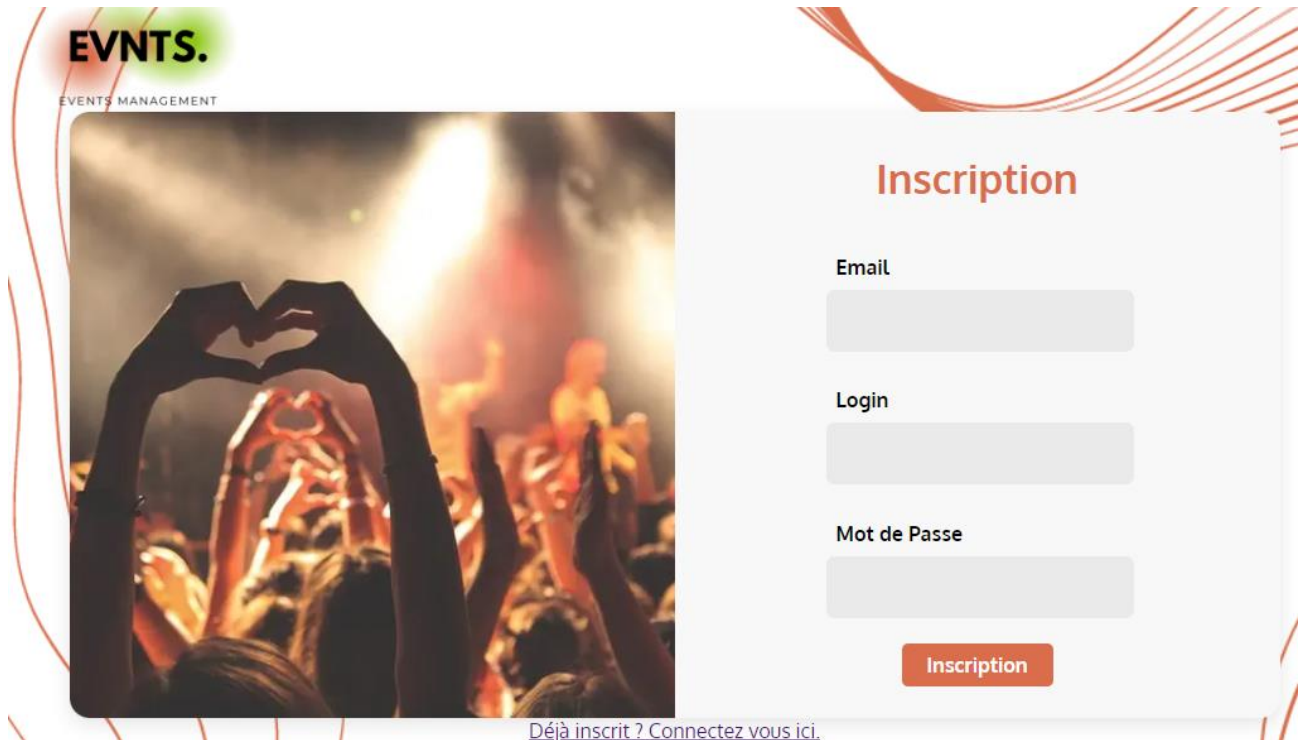
### III. Fonctionnalités opérationnelles



## 1. Service Authentification

Le service d'authentification permet à un utilisateur de l'application de créer un compte et de se connecter.

Une fois que celui-ci a créé son compte, ses informations sont transmises dans les BDD de tous les services. Après sa création, il peut se connecter afin d'accéder au contenu de l'application.



**EVNTS.**  
EVENTS MANAGEMENT

### Inscription

Email

Login

Mot de Passe

**Inscription**

[Déjà inscrit ? Connectez vous ici.](#)



**EVNTS.**  
EVENTS MANAGEMENT

### Connexion

Login

Mot de Passe

**Connexion**

[Pas encore inscrit ? Inscrivez vous ici.](#)

Les implémentations fonctionnelles :

- Inscription

- Connexion
- Déconnexion
- Modifier le pseudo
- Modifier le mot de passe
- Supprimer un utilisateur

## 2. Service Salon

Le service Salon est le plus gros webservice de l'application. Sa raison d'être est la création et la gestion de tous les salons et événements. Un utilisateur se connectant à l'application peut créer un salon, un événement. Il peut ensuite modifier les salons dont il est le créateur (ou modérateur) et ajouter/supprimer/modifier des événements. Un utilisateur peut envoyer des messages dans le salon où il est ou encore dans les événements auxquels il est inscrit.

Les implémentations fonctionnelles :

- Création d'un salon
- Modification d'un salon
- Récupération d'un salon par son ID
- Récupération d'un salon par son nom
- Récupération d'un utilisateur par son pseudo
- Modification d'un événement
- Mettre un utilisateur comme "présent" à un événement
- Mettre un utilisateur comme "absent" à un événement
- Créer un événement
- Récupérer un événement par son nom et son salon
- Valider un événement
- Envoyer un message dans un salon
- Envoyer un message dans un événement
- Récupérer les messages d'un salon
- Récupérer les messages d'un événement
- Rejoindre un salon
- Supprimer un utilisateur
- Récupérer un utilisateur
- Ajout un utilisateur en tant que modérateur
- Supprimer un modérateur
- Récupérer les salons d'un utilisateur
- Récupérer les événements d'un utilisateur
- Ajouter un membre (pour donner suite à l'inscription)
- Récupérer un utilisateur par son e-mail

Les implémentations non terminées :

- Inviter un utilisateur

### 3. Service GestionUtilisateur

Les implémentations fonctionnelles :

- Inscription de l'utilisateur (appel par l'authentification)
- Changer le pseudo d'un utilisateur
- Changer la bio d'un utilisateur
- Changer la photo de profil d'un utilisateur
- Supprimer un utilisateur
- Récupérer les informations d'un utilisateur
- Ajout d'un utilisateur dans la liste des contacts d'un autre

### 4. Service Contact

Le service Contact permet de pouvoir communiquer entre utilisateur. Il permet à un utilisateur de gérer ses contacts et ses conversations, en envoyant des messages, les consultants ou encore en ajoutant/retirant des contacts.

Pour ce service uniquement le back est fonctionnel. Nous n'avons pas réussi à l'ajouter au consul. Il y a eu beaucoup d'essais qu'on peut retrouver dans certains fichiers (Program.cs / Startup.cs). Il est cependant dockérisé.

Les implémentations fonctionnelles :

- Ajout d'un contact
- Ajout d'une conversation
- Création d'un utilisateur
- Demande de contact
- Envoie d'un message
- Récupérer les contacts d'un utilisateur
- Récupérer les conversations d'un utilisateur
- Récupérer les demandes d'ajout
- Supprimer un contact
- Supprimer un utilisateur
- Visionner un message

### 5. Service Google

L'objectif d'un de nos microservices était d'avoir la possibilité de se connecter à partir d'une authentification Gmail, et de proposer à partir de cela différents services.

Nous avons donc créé une boîte mail ([evntsinfo.orleans@gmail.com](mailto:evntsinfo.orleans@gmail.com)), et dans celle-ci nous avons créé un projet s'appuyant sur le Cloud de Google.

Google Cloud

Evnts

Tapez / pour rechercher des ressources, des documents, des produits, etc

Recherche

API et services

API et services activés

Bibliothèque

Identifiants

Écran de consentement OAuth

Page des accords d'utilisation

Identifiants

+ CRÉER DES IDENTIFIANTS

SUPPRIMER

RESTAURER DES IDENTIFIANTS SUPPRIMÉS

Créez des identifiants pour accéder à vos API activées. [En savoir plus](#)

Clés API

<input type="checkbox"/>	Nom	Date de création ↓	Restrictions	Actions
<input type="checkbox"/>	API key 1	6 avr. 2024	Aucune	<a href="#">AFFICHER LA CLÉ</a>

ID clients OAuth 2.0

<input type="checkbox"/>	Nom	Date de création ↓	Type	ID client	Actions
<input type="checkbox"/>	Test Google Auth	6 avr. 2024	Application Web	46735889023-rhhgq...	<a href="#">✎</a> <a href="#">🗑</a> <a href="#">⬇</a>
<input type="checkbox"/>	EvntsOAuthClient	6 avr. 2024	Application Web	46735889023-0vkg8...	<a href="#">✎</a> <a href="#">🗑</a> <a href="#">⬇</a>

Comptes de service

[Gérer les comptes de service](#)

<input type="checkbox"/>	E-mail	Nom ↑	Actions
Aucun compte de service à afficher			

Au sein du projet Java, un microservice gestionGoogle est créé, et est lié à l'API google dans le /ressources au sein du application.properties, où un client id lie l'ensemble.

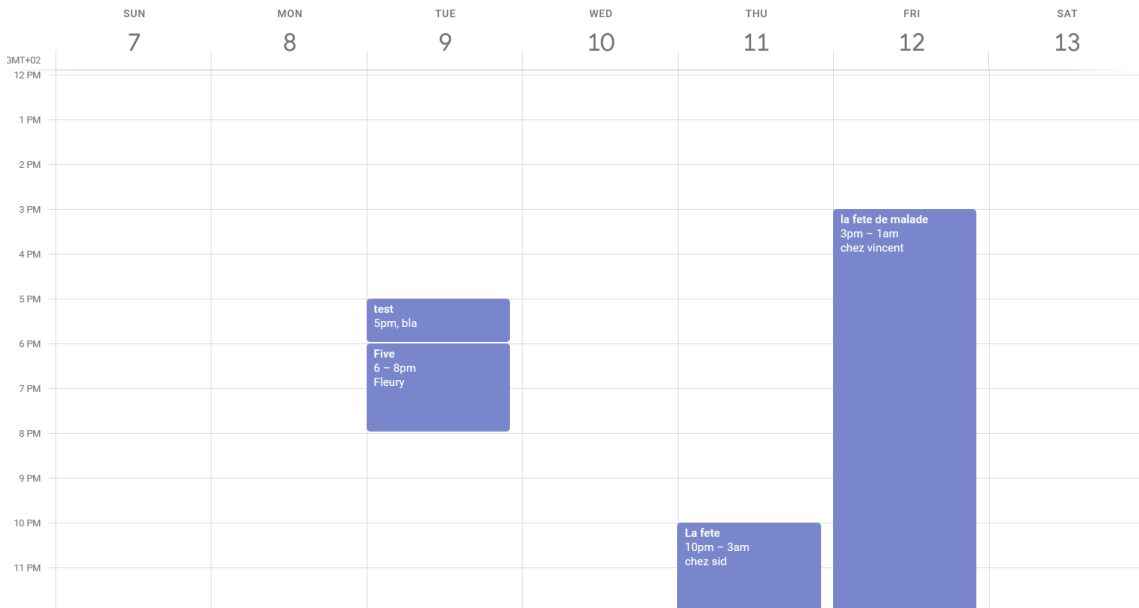
Nous n'avons pas eu le temps d'intégrer ce microservice au front du projet, il est donc toujours en phase de test, avec un front en HTML classique pour essayer les fonctionnalités en étant connecté.

Par ailleurs, comme notre application n'est pas une application publiée, l'application est en whitelist, donc la liste des personnes pouvant s'y connecter est limité.

Mais lier notre projet à l'API Google permet d'utiliser leurs services proposés, et en tant qu'application événement nous avons trouvé pertinent de pouvoir ajouter un événement au Calendar Google.

Ce microservice propose donc 2 solutions différentes au sujet du Google Calendar :

- La création directement d'un événement dans ce dernier, grâce au lien personnel de l'utilisateur connecté



- La génération d'une URL permettant lorsqu'il est copié dans la barre de recherche, de créer automatiquement un évènement dans le Calendar Google. Par exemple, cet URL :  
<https://calendar.google.com/calendar/u/0/r/eventedit?text=FinProjet&dates=20240421T200000Z/20240421T235900Z&ctz=Europe/Paris&details=cest%20la%20fin&location=orleans> possède dans son lien tout le nécessaire pour créer un événement dans son calendrier

× FinProjet Save

Apr 21, 2024 10:00pm to 1:59am Apr 22, 2024 (GMT+02:00) Central European Time - Paris Time zone

☐ All day Does not repeat

**Event details** Find a time

Add Google Meet video conferencing

orleans

Notification 30 minutes

Add notification

Sidney Gitteau

Busy Default visibility

cest la fin

**Guests**

Add guests

Guest permissions

☐ Modify event

☒ Invite others

☒ See guest list

Cela permettrait pour les personnes connectées via Google ou non de pouvoir ajouter un évènement dans son calendrier.

Créer une adresse mail à l'application permet aussi à notre API d'envoyer un mail. Deux méthodes sont aussi créées, l'une permettant d'envoyer un mail prédéfini en dur dans le code à l'utilisateur connecté, et une autre qui à partir d'un mail cible, un objet et un message d'envoyer un mail au destinataire.

Les implémentations fonctionnelles :

- Envoi de mail vers un destinataire depuis l'adresse mail de l'application
- Création d'un lien permettant l'ajout d'un événement dans le google calendar
- Création d'un utilisateur
- Suppression d'un utilisateur

Depuis l'environnement "recette" du microservice :

- Ajout d'un évènement directement dans le Google Calendar du client connecté

- Envoi de mail au client connecté
  - Récupération des évènements entre 2 dates données
- 

## Guide utilisateur

Le guide d'utilisateur est disponible dans le README.md du git.

<https://github.com/Baptie/evnts/blob/master/README.md>

L'intégralité du projet est disponible sur le git.

<https://github.com/Baptie/evnts>

Lien vers la vidéo de présentation :

<https://youtu.be/Hc602ETuVSc>