

Épreuve intégrée de la section

Bachelier en informatique de gestion

**ENSEIGNEMENT supérieur ÉCONOMIQUE de type court**

**Application Web   
Réseau social thème plongée**

**Enjoy Your Dive  
  
Année académique 2023-2024**

Travail de fin d'études présenté par **Benjamin Sterckx** en vue de l'obtention du diplôme de Bachelier en Informatique de Gestion

Directeurs de projet :

**M. Bernair  
M. Martin**

**Table des matières**

[1. Introduction 5](#__RefHeading___Toc880_1207985230)

[2. Contexte 5](#__RefHeading___Toc882_1207985230)

[2.1 Le client 5](#__RefHeading___Toc884_1207985230)

[2.2 La demande initiale 5](#__RefHeading___Toc886_1207985230)

[2.3 Les produits existants 5](#__RefHeading___Toc888_1207985230)

[3. Cahier des charges 5](#__RefHeading___Toc890_1207985230)

[3.1 Élaboration 5](#__RefHeading___Toc892_1207985230)

[3.1.1. Les interviews 5](#__RefHeading___Toc894_1207985230)

[3.1.2. Les demandes du client 6](#__RefHeading___Toc896_1207985230)

[3.1.3. Les propositions au client 6](#__RefHeading___Toc898_1207985230)

[3.2. Lots d’informations identifiés 6](#__RefHeading___Toc900_1207985230)

[3.2.1. Lots d’informations existants à conserver 6](#__RefHeading___Toc902_1207985230)

[3.2.2. Lots d’informations existants à remplacer 6](#__RefHeading___Toc904_1207985230)

[3.2.3. Lots d’informations à produire 6](#__RefHeading___Toc906_1207985230)

[3.3. Les acteurs de l’environnement d’exploitation de l’application 7](#__RefHeading___Toc908_1207985230)

[3.3.1. Le visiteur 7](#__RefHeading___Toc910_1207985230)

[3,3.2. Le plongeur 7](#__RefHeading___Toc912_1207985230)

[3.3.3. Le formateur 7](#__RefHeading___Toc914_1207985230)

[3.3.4. L’administrateur 7](#__RefHeading___Toc3334_1313685361)

[3.3.5. Le super administrateur 7](#__RefHeading___Toc918_1207985230)

[3.4. Définition de la première version de l’application 7](#__RefHeading___Toc920_1207985230)

[3.4.1. Les lots d’informations 7](#__RefHeading___Toc922_1207985230)

[3.4.2. Les fonctionnalités retenues 8](#__RefHeading___Toc924_1207985230)

[3.4.3. Scénario d’une session d’utilisation ordinaire 8](#__RefHeading___Toc926_1207985230)

[3.4.4. Infrastructure informatique 8](#__RefHeading___Toc928_1207985230)

[3.4.5. Validation de l’analyse 9](#__RefHeading___Toc930_1207985230)

[3.5. Perspectives d’évolution pour une version ultérieure 9](#__RefHeading___Toc2360_1207985230)

[4. Analyse 11](#__RefHeading___Toc940_1207985230)

[4.1. L’analyse des données 11](#__RefHeading___Toc942_1207985230)

[4.1.1. Modèle logique de données selon Mérise 11](#__RefHeading___Toc944_1207985230)

[4.1.2. Modèle conceptuel de données selon Mérise 12](#__RefHeading___Toc946_1207985230)

[4.2. L’analyse des traitements 12](#__RefHeading___Toc948_1207985230)

[4.2.1. Les Exigences fonctionnels 12](#__RefHeading___Toc950_1207985230)

[4.2.2. Les Exigences non fonctionnels 13](#__RefHeading___Toc952_1207985230)

[4.2.3. Les Exigences de performance 13](#__RefHeading___Toc954_1207985230)

[4.2.4 Les cas d’utilisation 14](#__RefHeading___Toc956_1207985230)

[4.2.5 Priorités et risques 14](#__RefHeading___Toc2362_1207985230)

[4.3. Les scénarios, diagrammes classe et de séquences 15](#__RefHeading___Toc2364_1207985230)

[4.3.1. Consulter 15](#__RefHeading___Toc2366_1207985230)

[4.3.2. Se connceter 16](#__RefHeading___Toc2368_1207985230)

[4.3.3. Gérer les évènement 19](#__RefHeading___Toc2370_1207985230)

[4.3.4. Gérer les messages entre utilisateurs 24](#__RefHeading___Toc2372_1207985230)

[4.3.5. Gérer son profil Description du scenario 28](#__RefHeading___Toc2374_1207985230)

[4.3.6. Gérer son carnet de plongée 32](#__RefHeading___Toc2376_1207985230)

[4.3.7. Gérer les clubs de plongée 37](#__RefHeading___Toc2378_1207985230)

[4.3.8. Gérer les sites de plongée 42](#__RefHeading___Toc2380_1207985230)

[4.3.9. Gérer les fédérations de plongée 47](#__RefHeading___Toc2382_1207985230)

[4.3.10. Gérer les droit 52](#__RefHeading___Toc2384_1207985230)

[4.4. Diagramme de packages 56](#__RefHeading___Toc2386_1207985230)

[4.5. Diagramme de déploiement 57](#__RefHeading___Toc958_1207985230)

[4.6. Choix des outils informatiques 58](#__RefHeading___Toc960_1207985230)

[4.6.1. La Base de données 58](#__RefHeading___Toc962_1207985230)

[4.6.2. Le Server 58](#__RefHeading___Toc964_1207985230)

[4.6.3. Le Client 58](#__RefHeading___Toc966_1207985230)

[4.6.4. Les interfaces de développement 58](#__RefHeading___Toc968_1207985230)

[4.7. Méthodoligie de travail utilisée 60](#__RefHeading___Toc960_12079852301)

[4.7.1 Mérise 60](#__RefHeading___Toc3290_1313685361)

[4.7.2 Scrum 60](#__RefHeading___Toc3292_1313685361)

[4.7.3 Agile 61](#__RefHeading___Toc3294_1313685361)

[4.7.4 CI/CD 61](#__RefHeading___Toc3296_1313685361)

[5. Réalisation 62](#__RefHeading___Toc970_1207985230)

[5.1. L’application 62](#__RefHeading___Toc972_1207985230)

[5.1.1 L’architecture 62](#__RefHeading___Toc3756_4065020134)

[5.1.2 L'installation 62](#__RefHeading___Toc3364_1313685361)

[5.1.3 La base de donnée 63](#__RefHeading___Toc3300_1313685361)

[5.1.3 Le server 64](#__RefHeading___Toc3425_1313685361)

[5.1.5 Le client 66](#__RefHeading___Toc3427_1313685361)

[5.2. Mise en service 69](#__RefHeading___Toc974_1207985230)

[5.2.1 Azure 69](#__RefHeading___Toc3308_1313685361)

[5.2.2 Les builds 69](#__RefHeading___Toc3310_1313685361)

[5.3. Problèmes rencontrés et des solutions trouvées 71](#__RefHeading___Toc976_1207985230)

[5.3.1 Le rafraichissement du client 71](#__RefHeading___Toc3312_1313685361)

[5.3.2 Le deployement 71](#__RefHeading___Toc3314_1313685361)

[5.3.3 La lenteur 71](#__RefHeading___Toc3326_1313685361)

[5.3.4 La latence 71](#__RefHeading___Toc3328_1313685361)

[5.3.5 L’utilisation d’une adresse SMTP 71](#__RefHeading___Toc3330_1313685361)

[5.3.6 Le temps 71](#__RefHeading___Toc3332_1313685361)

[5.4. Formation et suivi 72](#__RefHeading___Toc3316_1313685361)

[5.4.1 TechnofuturTic 72](#__RefHeading___Toc3318_1313685361)

[5.4.2 Transurb 72](#__RefHeading___Toc3320_1313685361)

[5.5. Le résultat 72](#__RefHeading___Toc3316_13136853611)

[5.5.1 Home 72](#__RefHeading___Toc3320_13136853611)

[5.5.2 S'enregister 72](#__RefHeading___Toc3320_131368536111)

[5.5.3 Connexion 73](#__RefHeading___Toc3320_1313685361111)

[5.5.4 Utilisateur connecté 73](#__RefHeading___Toc3320_131368536112)

[5.5.5 Profil 73](#__RefHeading___Toc3421_1313685361)

[5.5.6 Contacts 74](#__RefHeading___Toc3423_1313685361)

[5.5.7 Les sites 74](#__RefHeading___Toc3429_1313685361)

[5.5.8 Les groupes 75](#__RefHeading___Toc3431_1313685361)

[5.5.9 Les évènements 75](#__RefHeading___Toc3380_1313685361)

[5.5.10 les messages 76](#__RefHeading___Toc3382_1313685361)

[5.5.11 Admin 76](#__RefHeading___Toc3384_1313685361)

[6. Evolution(s) immédiate(s) pressentie(s) 79](#__RefHeading___Toc980_1207985230)

[6.1 Augementer les adherants 79](#__RefHeading___Toc3758_4065020134)

[6.2 Créer des nouvelles fonctionnalitées 79](#__RefHeading___Toc3760_4065020134)

[6.3 Monter une société 79](#__RefHeading___Toc3762_4065020134)

[6.4 Les ambassadeurs 79](#__RefHeading___Toc3764_4065020134)

[6.5 Un design 79](#__RefHeading___Toc3766_4065020134)

[6.6 Les invitations 79](#__RefHeading___Toc3768_4065020134)

[6.7 Les messages par groupes 79](#__RefHeading___Toc3770_4065020134)

[7. Évaluation du travail 79](#__RefHeading___Toc982_1207985230)

[7.1. Autocritique 79](#__RefHeading___Toc984_1207985230)

[7.1.1. L’élaboration du cahier des charges 79](#__RefHeading___Toc986_1207985230)

[7.1.2. La conduite de l’analyse 79](#__RefHeading___Toc988_1207985230)

[7.1.3. La réalisation 80](#__RefHeading___Toc990_1207985230)

[7.2. Appréciation(s) du client 80](#__RefHeading___Toc992_1207985230)

[8. Conclusion 80](#__RefHeading___Toc994_1207985230)

[9. Bibliographie 80](#__RefHeading___Toc996_1207985230)

[10. Annexes 80](#__RefHeading___Toc998_1207985230)

[10.1. Les scripts SQL 80](#__RefHeading___Toc1000_1207985230)

# **1. Introduction**

L’objectif est de créer une application style réseaux social sur le thème de la plongée en Belgique.  
Celle-ci proposera de mettre en relation les plongeurs de Belgique autour d’évènements, de groupes et d’une messagerie instantanée.  
Un partie administrateur permettra de gérer la modération de l’application  
Ce genre d’application n’existe pas encore, ce sera une nouveauté.   
Par ailleurs, la plongée étant ma passion, c’est un sujet que je connais bien et suis a même à développer une logique business adapté aux plongeurs toutes fédérations confondues

Lien vers l’application : <https://enjoyyourdive.azurewebsites.net/>

# 2. Contexte

L’application sera centré sur l’organisation d’évènements qu’il soit ludique ou formatif, la description de la pratique de la plongée des différentes fédérations et permettra à tous utilisateurs de se contacter facilement.  
Par la suite, l’application pourra évoluer en ajoutant de nouvelles fonctionnalités.

## 2.1 Le client

L’application est destinée à tous plongeurs ou personnes désirant se renseigner sur la plongée en Belgique.  
Les informations sur les événements ludique ou formation à venir, les différents site de plongées seront disponible sans créer de profil.  
Pour interagir dans l’application le client devra se créer un profil via l’onglet ‘s’enregistrer’.  
Dés lors, son compte restera actif pendant 24h. Au-delà de cette période, le client devra se connecter via l’onglet ‘se connecter’  
L’application est destiné à être libre d’accès et donc gratuit.

## 2.2 La demande initiale

* 1. Elle a pour but de répondre à une demande quant a l’organisation d’évènements de plongées que ce soit formatif ou ludique sans pour autant passer par les réseaux sociaux classique tel que Facebook.
  2. Chaque évènements reprendra les informations sur la validé des niveaux, assurance et certificat médical de chaque participant ce qui facilite grandement le travail de l’organisateur.  
     Toute demande de participation devra être validé par l’organisateur, celui-ci est responsable de ses évènements  
     Et enfin, l’application permettra à toutes personnes d’y découvrir la plongée en mettant contact tous les passionnées via la messagerie instantanée.

## 2.3 Les produits existants

Les plongeurs en Belgique utilise Facebook (évènement) mais pour cela ils doivent avoir un compte Facebook hors beaucoup d’entre eux aimerait ne plus être sur ce réseaux.  
Facebook étant un collecteurs de données a des fins lucratives, de moins en moins de gens sont en accord avec leur politique traitement de données. De plus Facebook devient, selon les utilisateurs interviewés, pollué par le publicité. En moyenne un post sur trois est une publicité.  
Par ailleurs Facebook n’est pas centrer sur la plongée, les informations sur la pratique s’y trouve mais elles sont difficile a trouver

# 3. Cahier des charges

## 3.1 Élaboration

### 3.1.1. Les interviews

Afin d’avoir un avis cohérent et pertinant, j’ai choisis d’interviewer 4 personnes ayant des profiles différents et qui serait susceptible d’utiliser l’application par la suite.

1. Le plongeur :   
Pour ce profil, j’ai fait appel a l’un de mes amis qui organise régulièrement des plongées pour son groupe de plongeur.   
2. L’instructeur SSI :  
C’est un professionnel de la plongée, il fait partie de l’organisation SSI qui est reconnue dans le monde entier   
Il organise des formations régulièrement.

3. L’instructeur PADI :

C’est un professionnel de la plongée, il fait partie de l’organisation PADI qui est reconnue dans le monde entier   
Il organise des formations régulièrement.

4. Le Course director PADI :  
C’est un professionnel de la plongée, il fait partie de l’organisation PADI qui est reconnue dans le monde entier   
Il forme les instructeur PADI

### 3.1.2. Les demandes du client

Les clients dans ce projet sont les 4 personnes interviewées. Par la suite les client seront au travers d’une évolution continue les utilisateurs de l’application.  
La mise en place d’un système d’ambassadeur est a l’étude. Cela consiste à demandé des retour régulier aux organisateurs d’événements en échange d’une meilleur visibilité de leur évènements, il seront mis en haut et en évidence dans la liste.

### 3.1.3. Les propositions au client

Les retour d’interview m’ont permis de mieux cibler les demandes comme la mise en place de validation de niveau, assurance et certificat par les administrateurs et un accès direct à messagerie en cas de question de ce type   
La gestion du carnet a également été revue afin que les participants à une même plongée puisse déterminer leur binômes (personnes ayant fait le même type, profil de plongée)

## 3.2. Lots d’informations identifiés

### 3.2.1. Lots d’informations existants à conserver

Une recherche a été faite afin de recenser les informations sur les différents site de plongée en Belgique.  
Celle-ci a permis d’implémenter une partie des sites afin que les premiers utilisateur aient déjà quelques infos.  
Par la suite, la modification ou nouveaux sites a implémenter sera fait par les administrateurs.  
\*\* ajouter les liens

### 3.2.2. Lots d’informations existants à remplacer

Les utilisateur pourront modifier ou supprimer les évènements et groupes, validé la participation à leur événements ou groupes   
Les administrateur pourront eux ajouter et modifier des organismes, formations et sites de plongée

### 3.2.3. Lots d’informations à produire

Les évènements

Les événements créés par les utilisateurs sont caractérisés par leur nom, la date à laquelle ils auront lieu et le lieu.

Si l’utilisateur possède un groupe, il pourra le lier à l’événement. De cette manière, l’ensemble du groupe y sera convié.

Si l’utilisateur possède un niveau instructeur, il pourra en faire une formation.

Les formations

Ils s'agit d'événements créés par des utilisateurs qui ont un niveau instructeur ou équivalent.

Ces instructeurs ne pourront créer de formations que pour les organismes dont ils sont certifiés 'instructeur' ou équivalent. Pour cela, l’organisme devra être référencé par un administrateur.

Les groupes

Ils sont créés par les utilisateurs et sont caractérisés par leur nom et éventuellement leur adresse. L’adresse n’est pas obligatoire, elle permet aux autres utilisateurs de savoir de quelle région est le groupe. Seule la commune est visible dans l’application.

Les sites

Ils sont créés et modifiés par les administrateurs mais sont visibles de tous.

Les sites sont caractérisés par leur nom, une image, une carte, une description, les coordonnées GPS, une URL vers le site officiel, la profondeur maximum du site, le prix d’entrée, s'il y a un compresseur pour gonfler les bouteilles, s'il y a la possibilité de se restaurer, et leur adresse.

Les utilisateurs connectés auront la possibilité de donner une note de 1 à 5 pour chaque site. Ils pourront voir leur note ainsi que la note moyenne de tous les utilisateurs.

Un bouton 'Go' pour ouvrir Google Maps afin de s'y rendre sera disponible.

Un bouton 'Url' pour ouvrir le site officiel sera également disponible.

Les organismes et formations

Ils sont créés et modifiés par les administrateurs.

Les organismes sont caractérisés par leur nom, une image, leur adresse.

Les formations sont caractérisés par leur nom, une image, une description, si c’est une spécialité ou non, l’organisme dont il appartienne et leur prérequis qui est la formation requise la commencer

## 3.3. Les acteurs de l’environnement d’exploitation de l’application

Les acteurs de l’application seront répertorié sur 4 niveaux/droits

### 3.3.1. Le visiteur

Aucunes inscriptions nécessaires, il aura accès aux différentes informations mais ne pourra s’inscrire à aucuns évènements ni ne pourra faire des demandes d’amis ni même implémenter son carnet de plongée. Il peut juste naviguer

### 3,3.2. Le plongeur

Il devra s’inscrire donc notifier : son nom, prénom, e-mail, son mot de passe.  
Il pourra dés lors s’inscrire aux évènements créer, créer lui même des événements si il a au minimum un niveau de plongeur autonome, implémenter son carnet de plongée, modifier son profil en y ajoutant les image/photo de son niveau les plus élevés, assurance, certificat et ajouter ses différents niveaux acquis.  
Il pourra également se créer des groupes afin de ressembler ses amis plongeurs et les modifier

### 3.3.3. Le formateur

Il aura les mêmes droits que le plongeur mais pourra créer des évènements de type formations liés à sa fédérations et implémenter la partie groupes.  
Le formateur peut faire une demande afin de devenir administrateur et gérer cette partie.

### 3.3.4. L’administrateur

Il aura les mêmes droits que le formateur mais pourra implémenter l’application de nouvelles données a savoir : nouveaux sites de plongées, nouvelles fédérations, nouvelles formation ou spécialités .   
Il pourra également bloquer des utilisateur en cas d’abus, de non respect des règles de l’application

### 3.3.5. Le super administrateur

Il aura les mêmes droits que l’administrateur et lui pourra donner ou enlever les droits administrateur.

## 3.4. Définition de la première version de l’application

### 3.4.1. Les lots d’informations

L’application se composera donc dans l’ordre d’implémentation  
  
1.Page de garde avec son logo

2. Menu de navigation générale dans lequel on retrouve

Events: ce sont tous les événements qui ne sont pas des formations

Trainings: ce sont tous les événements qui sont des formations

Groupes: ce sont tous les groupes

Sites: ce sont tous les sites de plongées  
 Contacts: ce sont les statu des personnes qui désirent rentré en contact avec nous et créer des lien d’amitié

Inscription: c’est le formulaire d’inscription

Connexion: c’est le formulaire pour se connecter

3. Menu de navigation personnel dans lequel on retrouve :  
 Image et nom: Profil utilisateur, reprend toutes les infos et permet d’y ajouter des images et niveaux  
 mon carnet: ce sont tous nos événements passé. On peut y ajouter des informations via ajouter au carnet

mes events: ce sont tous les événements ou je participes.

mes groupes: ce sont tous les groupes ou je participes.

messages: permet d’accéder aux messages de mes amis  
 admin: permet d’ouvrir le menu de navigation administrateur

4. Menu de navigation administrateur dans lequel on retrouve :

Validation: permet de sélectionner l’utilisateur afin le valider ses niveau, assurance et certificat

Formations: permet de créer ou modifier de nouvelles organisations et formations qui seront référencés.

Droit: permet de ajouter au supprimer les droits administrateur

Site: permet d’ajouter ou modifier des sites de plongées référencés dans l’application

5. Les formulaires

De connexion, d’enregistrement et de changement de mot de passe, mais encore  
Création et modification

- d’évènements

- de groupes

- d’organismes

- de formations

- d’adresse

- de carnet

- de niveau

- de profil

### 3.4.2. Les fonctionnalités retenues

S’inscrire  
Se connecter  
Gérer son profil  
Participer à des évènements

Gérer des évènements  
Gérer des groupes  
Gérer des organismes et formations  
Gérer des sites de plongées  
Gérer son carnet de plongée  
Coter les sites de plongée

Gérer les liens d’amitié  
Gérer les messages instantanés  
Ajouter/Supprimer les droits ( uniquement le super administrateur)

### 3.4.3. Scénario d’une session d’utilisation ordinaire

L’utilisateur se connecte avec son e-mail et son mot de passe   
En fonction de ses droits, il aura accès ou non a certaines parties de l’application  
Prenons exemple de l’utilisateur ayant de plus de droit l’administrateur.   
Il pourra naviguer sur l’ensemble de l’application et accéder a toutes les fonctionnalités décrient au point 3.4.2

### 3.4.4. Infrastructure informatique

L’application aura 3 parties distinctes qui elles mêmes seront découpées en plusieurs couches

3.4.4.1. Une base de données.

Elle sera mise en place grâce à SQLServer .

L’analyse ce cette base de donnée sera faite grâce Merise et 2 schéma seront créés, le modèle conceptuel des données et le modèle logique des données.   
Les schéma seront réalisés grâce au logiciel Looping

Dés lors tables et leurs liens par clés étranger seront créer

3.4.4.2. Une API

Elle sera codé en C# avec le framework .NET6. et sera composée de 3 couches   
B.1. Data Access Layer: elle permet d’accéder aux données de la base de données

B.2. Buisness Logic Layer: elle permet de créer les objets business de l’application  
B.3. Controllers: elle permet la communications entre l’API et le client via des route RESTFULL et permettra de sécurisé celle-ci grâce a un Token généré lors de chaque connexion

3.4.4.3. Un client

Codé avec Angular, c’est un framework front-end codé en TypeScript de type single page application  
Comportera plusieurs dossiers tels que  
  
Les Services   
- Http permet l’accès a l’API via des requêtes HTTP

- CustomValidator pour faire de la validation de formulaire non repris par le framework

- Guard pour controler l’acces aux urls  
- Helper, convertisseur de données

- Interceptor permet d’intercpter une requete entrante ou sortant afin d’y ajouter un logique

- Modal permet de gerer l’ouverture, la fermeture et les données des modals

- Navigation permet de mémorisé les pages consultés

- Resolver permet d’exécuter un code logique lors d’un navigation

- Session permet de formater et mémorisé les données utilisateur

- Observable permet de lire des données et leur modification en temps réel  
  
Les Composants

Ce sont des parties de vues, nous y retrouvons pour chacun d’entre eux : 1 fichier .html, 1 fichier .scss et1 fichier .ts

La partie Composants comportera plusieurs dossiers  tels que:

- Bars: ce sont les barres de navigation

- Cores: c’est le resultat de la naviation, les diffenrents vue

- Modals: partie de vue afficher en pop up

- User: vue dédié a l’utilisateur

- View-parts: portion de vue utiliser dans plusieurs autres vues

Environnements

reprend l’url pour acceder la l’API  
  
Models  
- les formulaires: objets du framwork permettant une gestion facile  
- les interfaces: objets représentant les données de l’application  
  
Routing

Permet de naviguer via des path et de les sécuriser

Module

Permet charger tout les éléments citer ci dessus dans le framewok

Nodes Modules

Ce sont les dépendances nécessaire aux bon fonctionnement de l’application

### 3.4.5. Validation de l’analyse

Aucunes remarques a ce jour n’a été émient par les personnes interviewées

## 3.5. Perspectives d’évolution pour une version ultérieure

Plusieurs améliorations sont à l’étude, telles que :

- Référencer des sites de plongées dans d’autres pays

- Implémenter d’autres fédérations de plongées connues dans le monde

- Financer l’application par des sponsors.

- Rendre l’application encore et toujours plus rapide.

- Obtenir les retours des utilisateurs via des sondages ou des formulaires.

- Permettre d’inviter des utilisateurs aux événements que j’ai créés.

- Créer des filtres sur la page ‘Events’, ‘Training’, ‘Groupes’ afin de cibler plus facilement l’élément recherché.

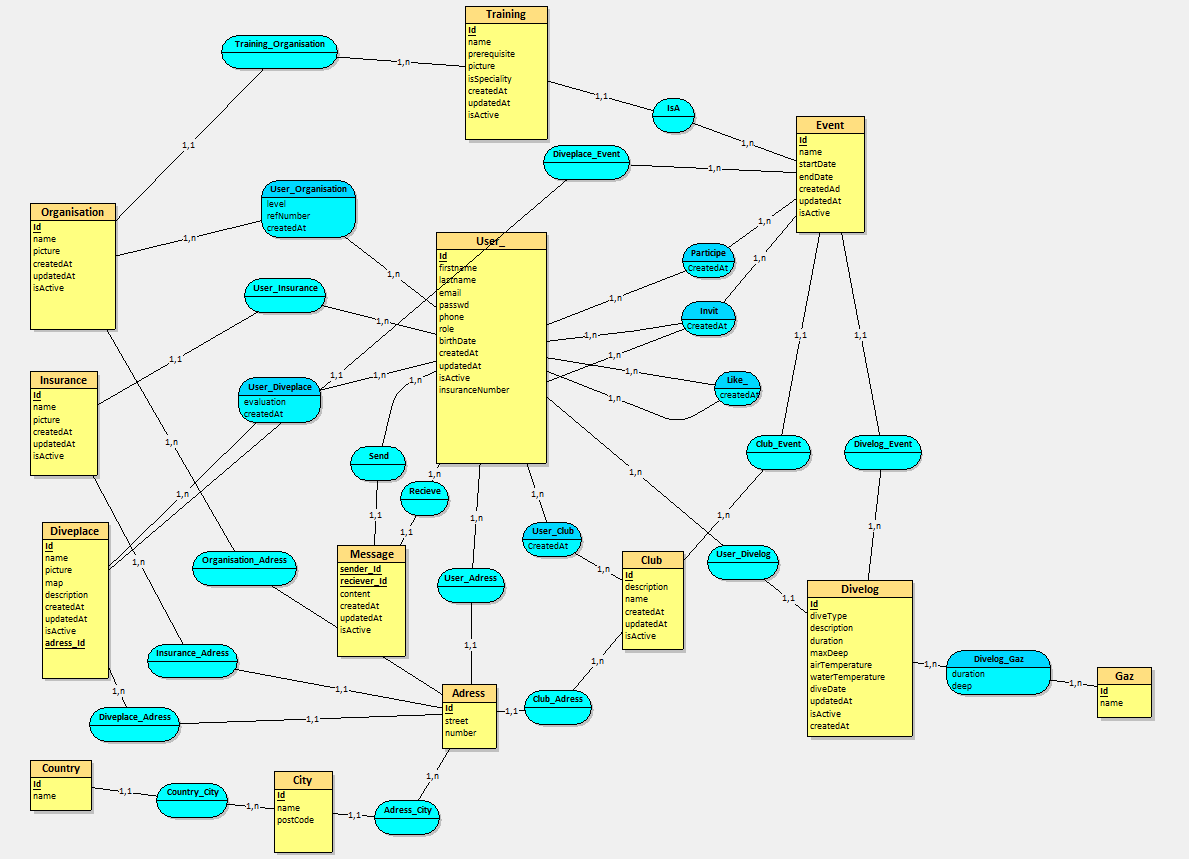
- Créer un système de notation pour les ambassadeurs. Ceux ci, grâce a leur participation, à la promotion et l’évolution recevraient un note qui mettrait en avant leur publication

Cette liste s’agrandit de jour en jour.

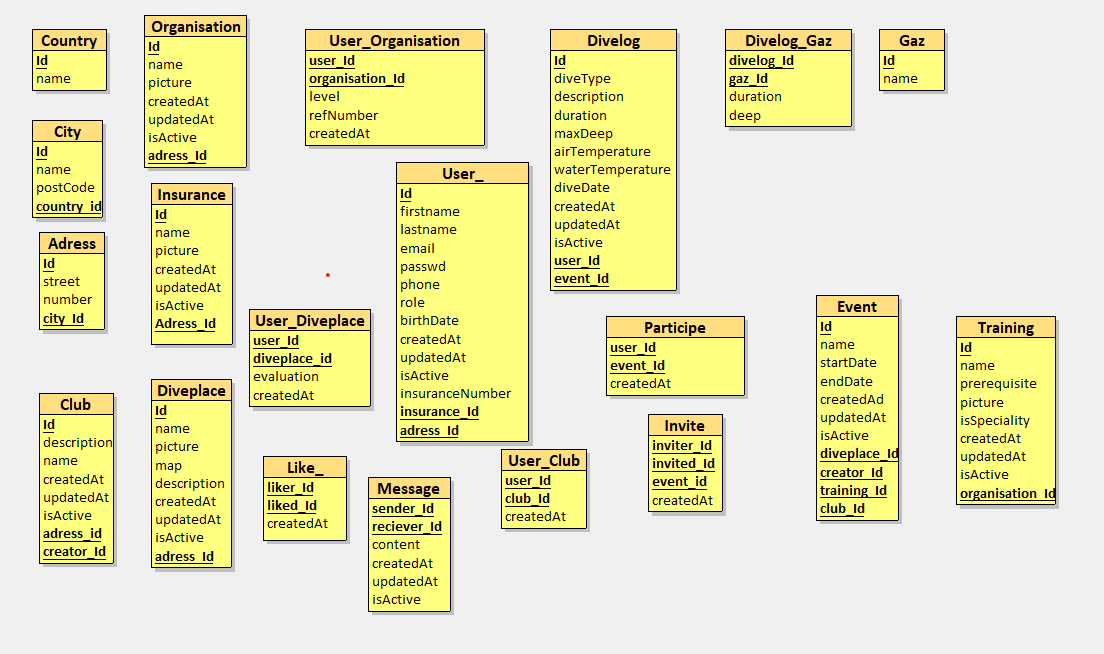
# 4. Analyse

## 4.1. L’analyse des données

### 4.1.1. Modèle logique de données selon Mérise



### 4.1.2. Modèle conceptuel de données selon Mérise



## 4.2. L’analyse des traitements

### 4.2.1. Les Exigences fonctionnels

4.2.1.1. Hors connexion

Pouvoir se créer un profil via s’enregistrer  
Pour se connecter via se connecter  
Consulter les Events, Trainings, Groupes, Sites et leur différentes informations sans pouvoir interagir avec ceux-ci

4.2.1.2. Connecter

Consulter les Events, Trainings, Groupes, Sites, leur différentes informations et interagir avec ceux-ci

Permettre à chaque utilisateur d’avoir le contrôle sur ses événements et groupes au travers de création, modification et suppression de ceux-ci

Faire une demande de participation à un évènement via un pouce levé  
Supprimer cette demande via un pouce baissé

Valider la demande si c’est un de nos évènement grâce au bouton ‘+‘

Enlever cette demande si c’est un de nos évènement via un bouton ‘-’

Supprimer les participant a nos évènements dans le formulaire modification

Faire une demande de participation à un groupe via un pouce levé  
Supprimer cette demande via un pouce baissé

Valider la demande si c’est un de nos groupe grâce au bouton ‘+‘

Enlever cette demande si c’est un de nos groupe via un bouton ‘-’

Supprimer les participants a nos groupes dans le formulaire modification

Voir dans chaque événements ou groupes l’age, le statu médical,le statu d’assurance par un V pour valide ou un X pour non valide, le niveau validé de chaque utilisateur via un bouton ‘œil’ et un accès conversation via un bouton ‘bulle de discutions’.  
  
Dans le carnet, permettre d’ajouter ou de modifier des informations sur la plongée effectuée dans notre carnet via un bouton ‘Ajouter +’

Dans Contacts, créer des liens d’amitié avec la validation des 2 utilisateurs via un bouton ‘smiley’

Enlever cette demande via un bouton ‘smiley fâché’

Supprimer cette demande via un bouton ‘poubelle’  
Voir les demandes en attende et les demandes reçues

Voir tous les contacts possible afin de leur faire une demande

Voir tous les amis   
Lors de lien d’amitié ceux ci seront directement accessible via la messagerie.

Dans la navigation Admin:   
Dans Validation, gérer la validation du statu médical, d’assurance de chaque utilisateur en affichant leur images et en encodant la date de fin de validation.  
Dans cette page choisir le niveau à afficher pour utilisateur en appuyant le un bouton ‘œil’

Dans Formation, gérer la création, modification et suppression des organismes et formations.  
Dans Droit, valider les droits admin avec le bouton ‘smiley lunette’, les enlever avec le butons ‘smiley’

Danse Site, gérer la création, modification et suppression des organismes et formations.

### 4.2.2. Les Exigences non fonctionnels

- Ergonomie sobre calquer sur ce que fait Facebook

- Design utilisation de Boostrap 5.0, des flex box, de Boostrap Icons

- Formulaire aide au remplissage via des avec message d’erreurs et validateurs

- Lecture fluide

### 4.2.3. Les Exigences de performance

- Gérer un nombre d’utilisateurs simultanés: 1000

- Taille de la base de données adaptées: 10 gigas

- Gérer la latence entre les requêtes afin d’éviter les bugs liées aux multiples clics

### 4.2.4 Les cas d’utilisation

### 4.2.5 Priorités et risques

|  |  |  |  |
| --- | --- | --- | --- |
| Cas d’utilisation | Priorité | Risque | Itération |
| Consulter | Haute | Faible | 3 |
| S'inscrire | Haute | Fort | 2 |
| Se connecter | Haute | Fort | 4 |
| Gérer profil | Haute | Moyen | 3 |
| Gérer carnets | Faible | Moyen | 3 |
| Gérer évènements | Haute | Moyen | 6 |
| Gérer messages | Haute | Moyen | 4 |
| Gérer sites | Moyenne | Faible | 3 |
| Gérer fédérations | Moyenne | Faible | 3 |
| Gérer clubs | Moyenne | Faible | 3 |

## 4.3. Les scénarios, diagrammes classe et de séquences

### 4.3.1. Consulter

Description du scenario

Le visiteur clic sur les liens pour naviguer dans l’application

- Events

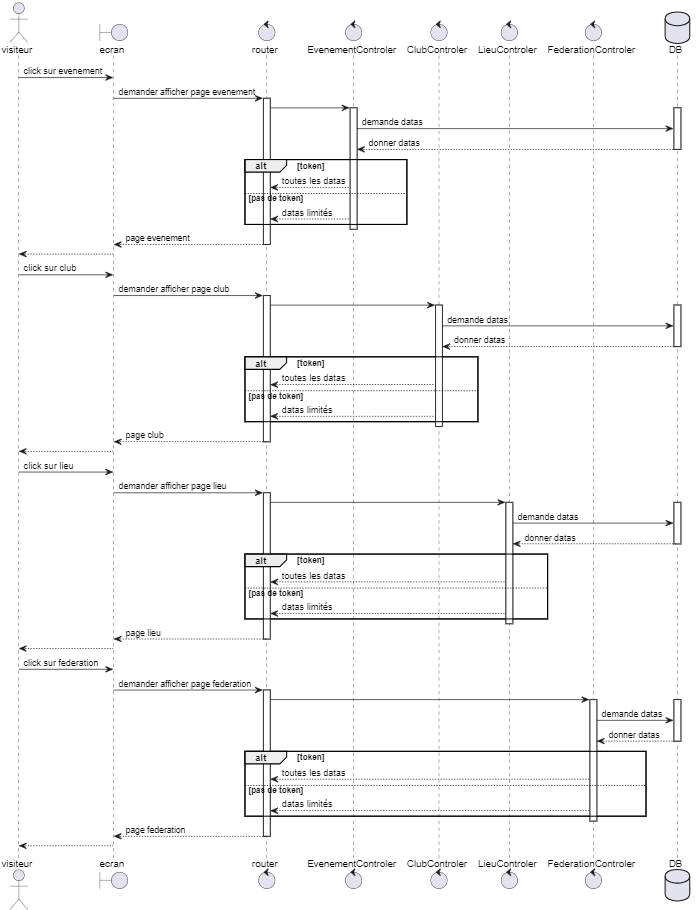
- Trainings

- Groupes

- Sites

|  |  |
| --- | --- |
| 1. Ce cas d’utilisation commence quand le visiteur arrive dans la page home | 2. Visuel d’un descriptif de l’application et accès au menu de navigation |
| 3. Clic sur l’un des liens suivants - évènements - clubs - lieux - fédérations | 4 .Navigue vers la page correspondante. Les infos sur la page sont limité. |

Diagramme de séquences



### 4.3.2. Se connceter

Description du scenario

Un visiteur arrive sur l’application et clic sur le liens s’inscrire

|  |  |
| --- | --- |
| 1. Ce cas d’utilisation commence quand l’utilisateur arrive dans la page home | 2. Visuel d’un descriptif de l’application et accès au menu de navigation |
| 3. Il clic sur s’inscrire | 4. Navigue vers la page inscription ou il y a un formulaire |
| 5. Il remplit au minimum les champ requit et valide | 6. Vérifications des champs puis enregistrement en base de donnée suivit du renvois d’un profil utilisateur avec création d’un Token |
| 7. Le visiteur devient utilisateur |  |

Diagramme de séquences

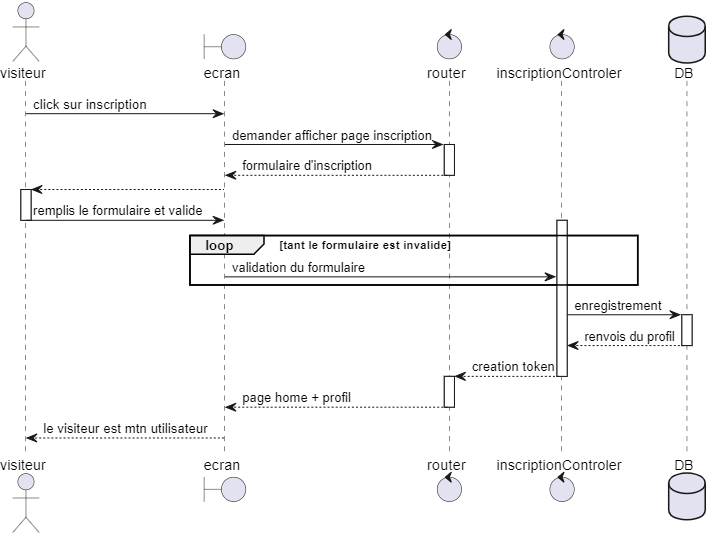


Diagramme de classe participatives

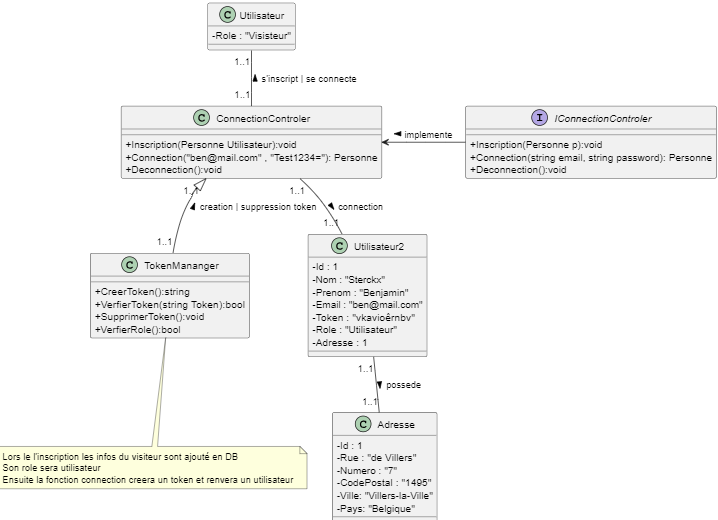
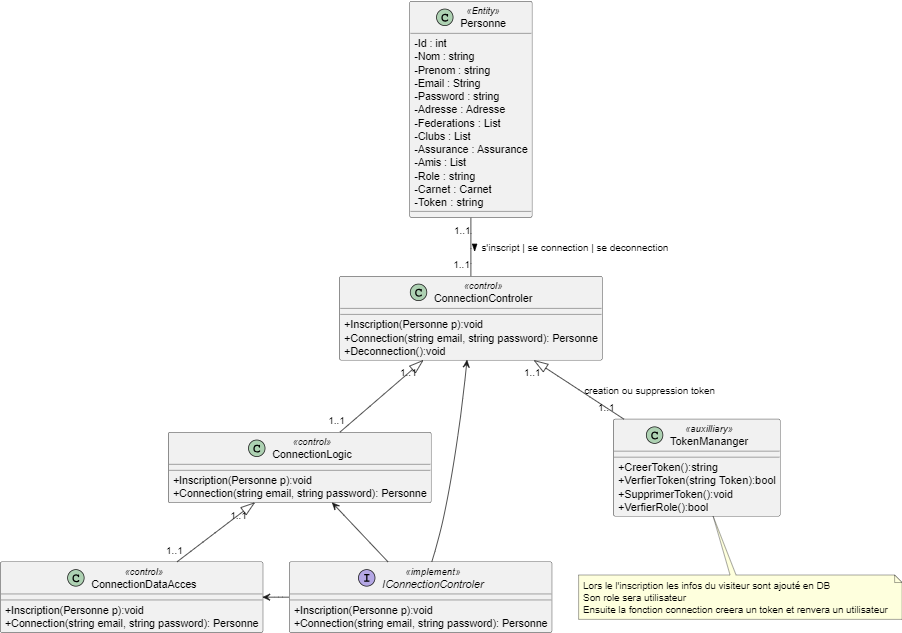


Diagramme de classes



### 4.3.3. Gérer les évènement

Description du scenario

L’utilisateur va sur la page évènement et clic sur le bouton créer, rempli un formulaire puis valide

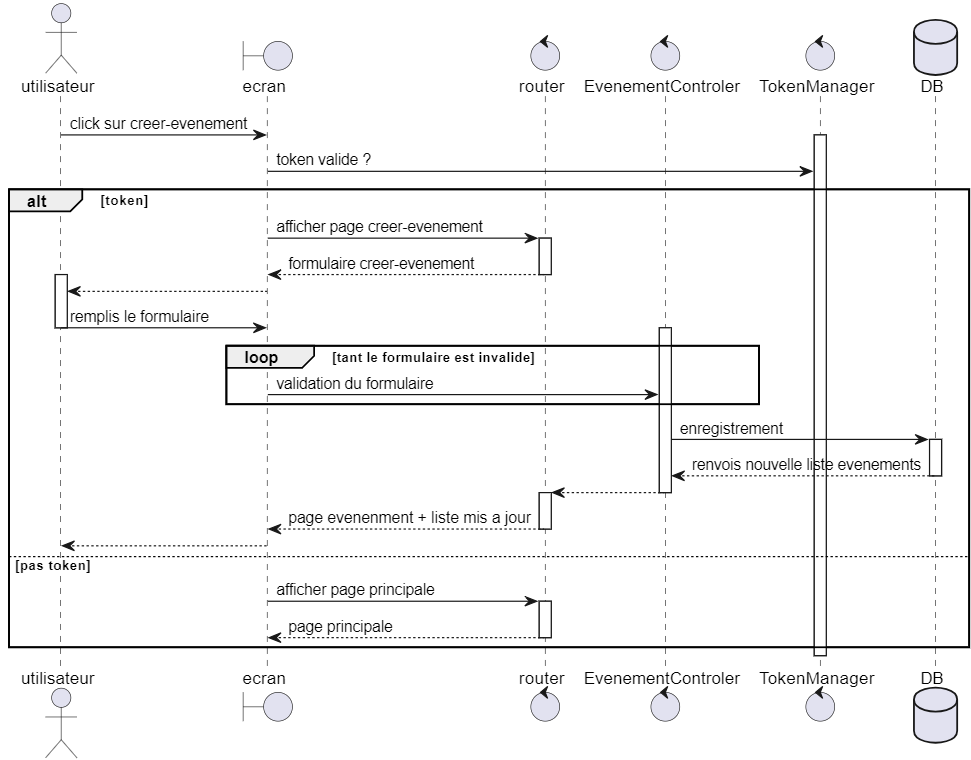
une fois l’évènement créer il pourra l’éditer, le désactiver ( rendre invisible) ou le réactivé

il peut inviter des plongeurs à son évènement

il peut participer au évènements auxquels il est convié

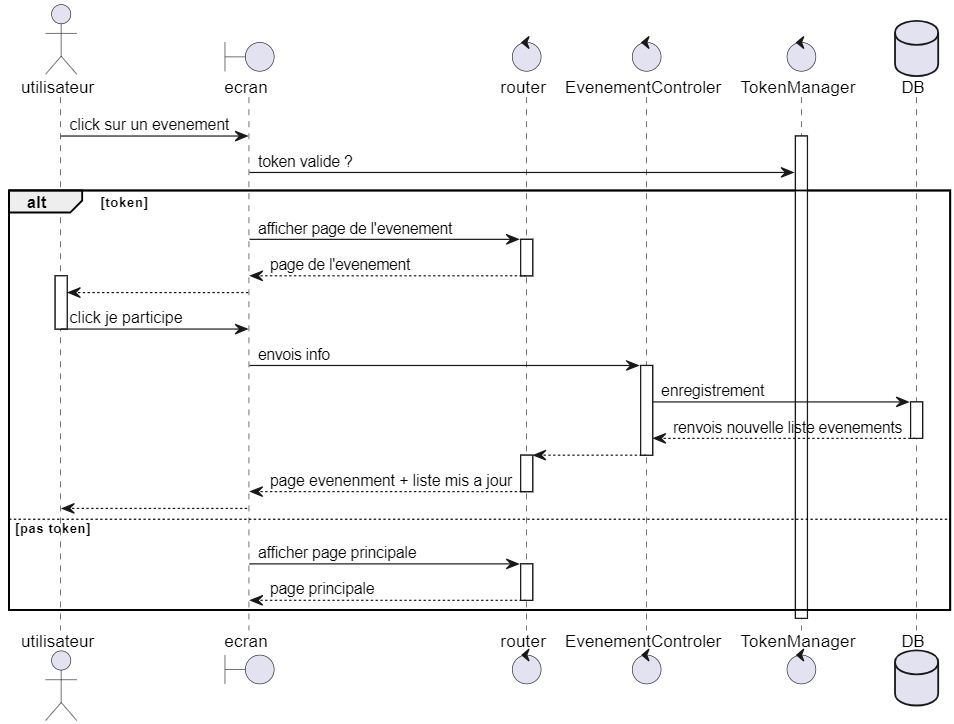
|  |  |
| --- | --- |
| 1. L’utilisateur navigue sur la page évènement | 2. la page évènement contient une liste  d’évènements créer par les membre de son club et  une liste d’évènement créer par ses amis |
| 3. L’utilisateur clic sur créer | 4 . si il est connecter, il est rediriger vers la page  creer-evenement  on récupère les infos :  - amis  - clubs  pour les afficher dans le formulaire  sinon il est redirigé vers la page principal |
| 5. il remplit le formulaire et valide | 6. Vérifications des champs puis enregistrement en base de donnée Elles seront accessible par les profiles concernés Il sera ensuite redirigé vers la page évènement |
| 7. il peut modifier, déactiver, réactiver ses évènements | 8. les donnes sont modifié en base de donnée et seront accessible par les profiles concernés Il sera ensuite redirigé vers la page évènement |

Diagramme de séquences



Scenario alternatif et Diagramme de séquences alternatif

|  |  |
| --- | --- |
| 1. L’utilisateur navigue sur la page évènement | 2. la page évènement contient une liste d’évènements créer par les membre de son club et une liste d’évènement créer par ses amis |
| 3. il peut choisir de participer a un évènement. clic sur je participe | 4. L’info est mémorisé dans la base de donnée et sera accessible par les profiles concernés |



|  |  |
| --- | --- |
| 1. L’utilisateur navigue sur la page évènement | 2. la page évènement contient une liste d’évènements créer par les membre de son club et une liste d’évènement créer par ses amis |
| 3. il peut choisir de inviter des amis a un évènement. choisis les amis dans la liste puis valide | 4. L’info est mémorisé dans la base de donnée et sera accessible par les profiles concernés |

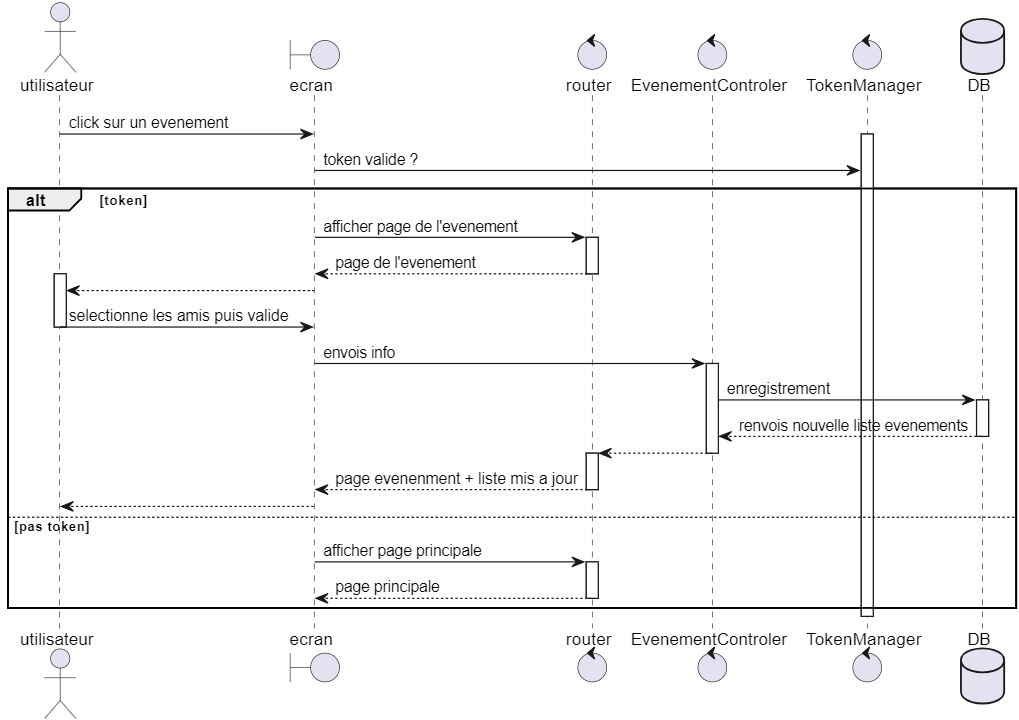


Diagramme de classe participatives

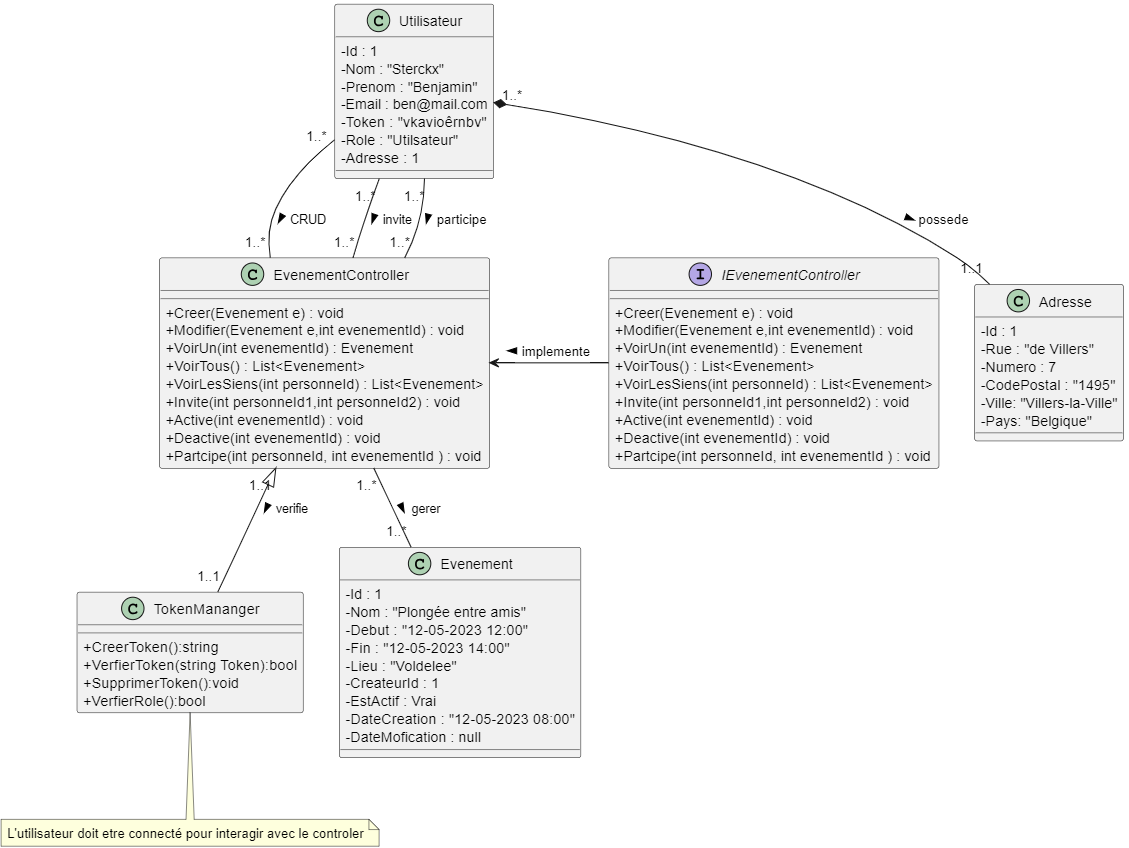
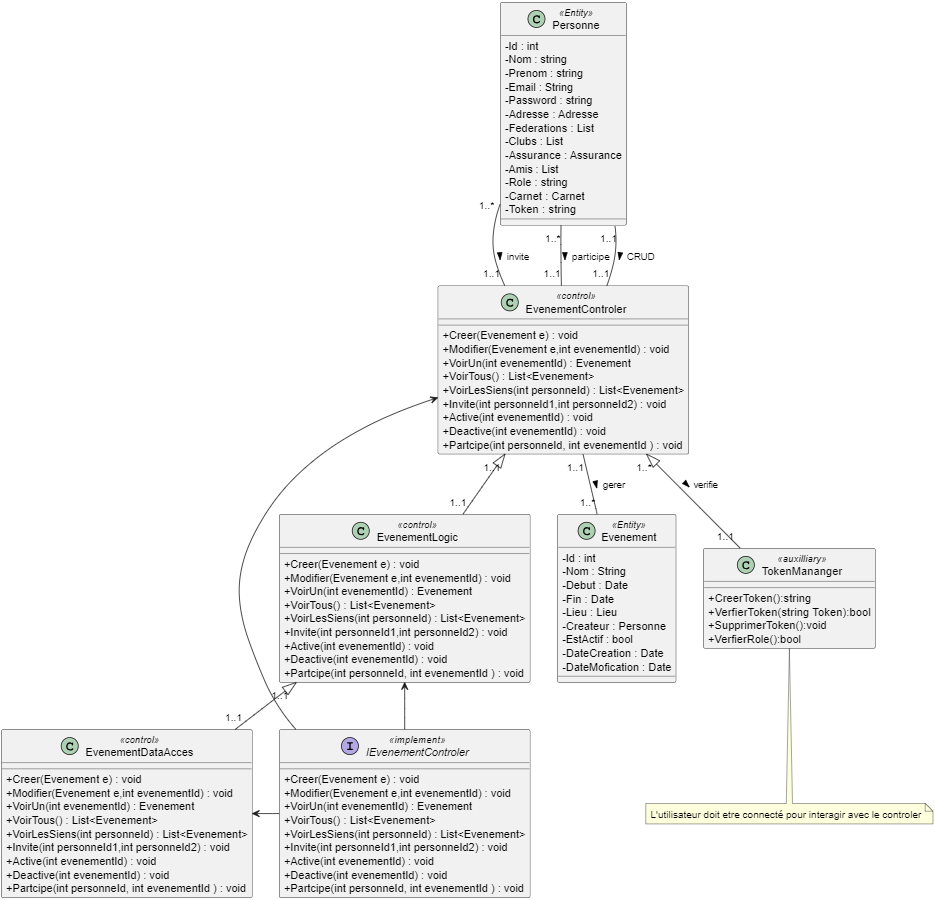


Diagramme de classes



### 4.3.4. Gérer les messages entre utilisateurs

Description du scenario

Une liste des utilisateur est afficher en haut a droit de l’écran.

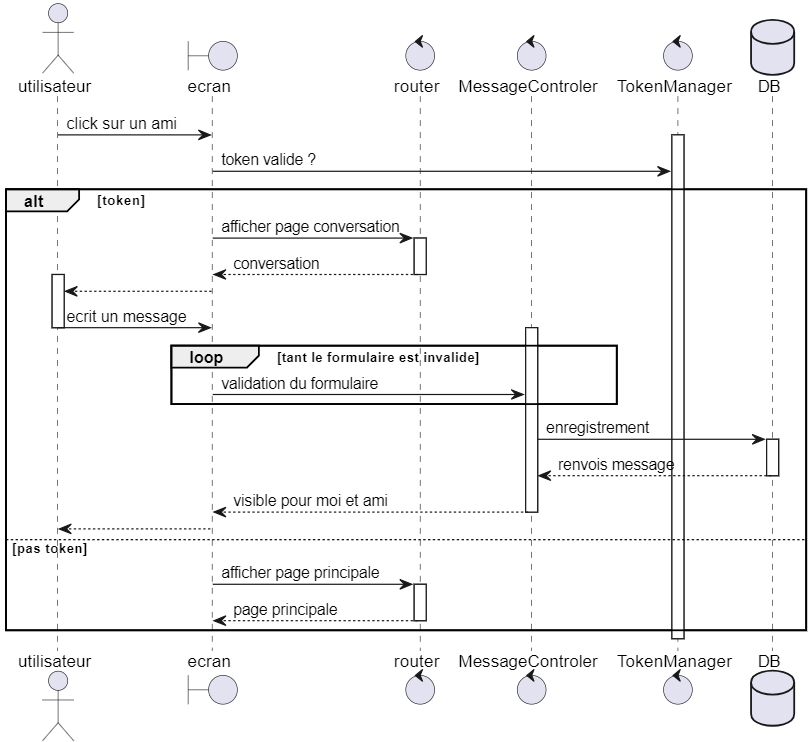
Ceux-ci sont triés par club.

L’utilisateur clic sur l’un entre eux et est rediriger vers la page de leur conversation

Il peut des lors envoyer, ou modifier des messages

|  |  |
| --- | --- |
| 1. L’utilisateur clic sur un autre utilisateur | 2. Redirection vers leur conversion |
| 3. il peut envoyer des messages | 4. L’info est mémorisé dans la base de donnée et  sera accessible par son ami |

Diagramme de séquences



Scenario alternatif et Diagramme de séquences alternatif

|  |  |
| --- | --- |
| 1. L’utilisateur clic sur autre utilisateur | 2. Redirection vers leur conversion |
| 3. Il peut modifier des messages | 4. L’info est mémorisé dans la base de donnée et sera accessible par son ami |

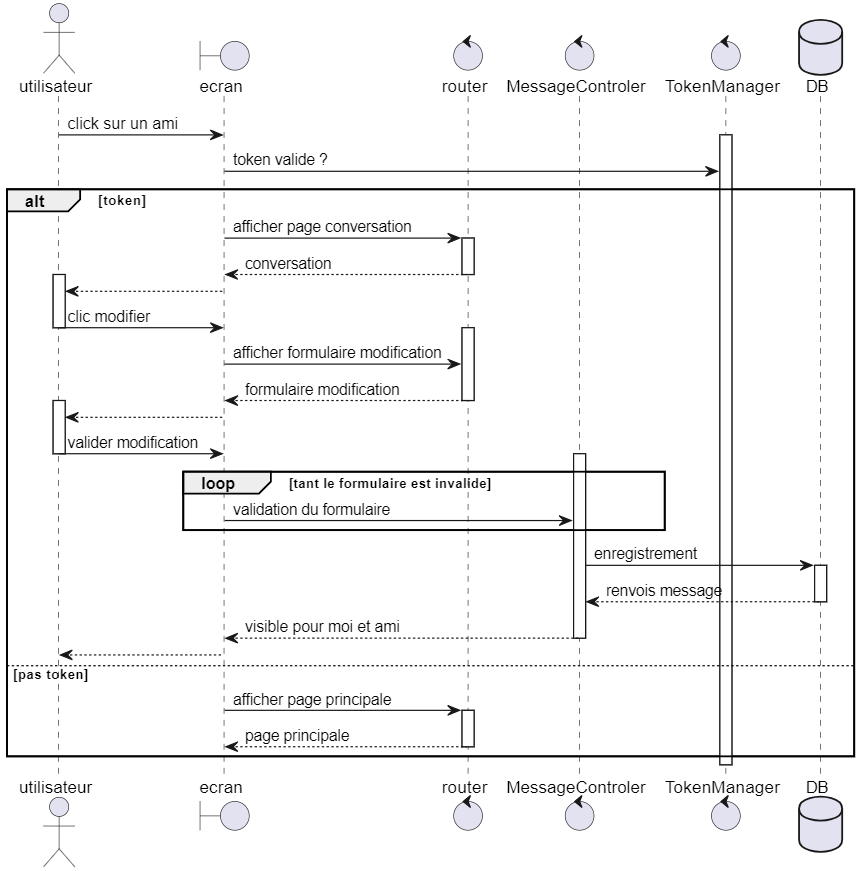


Diagramme de classes participatives

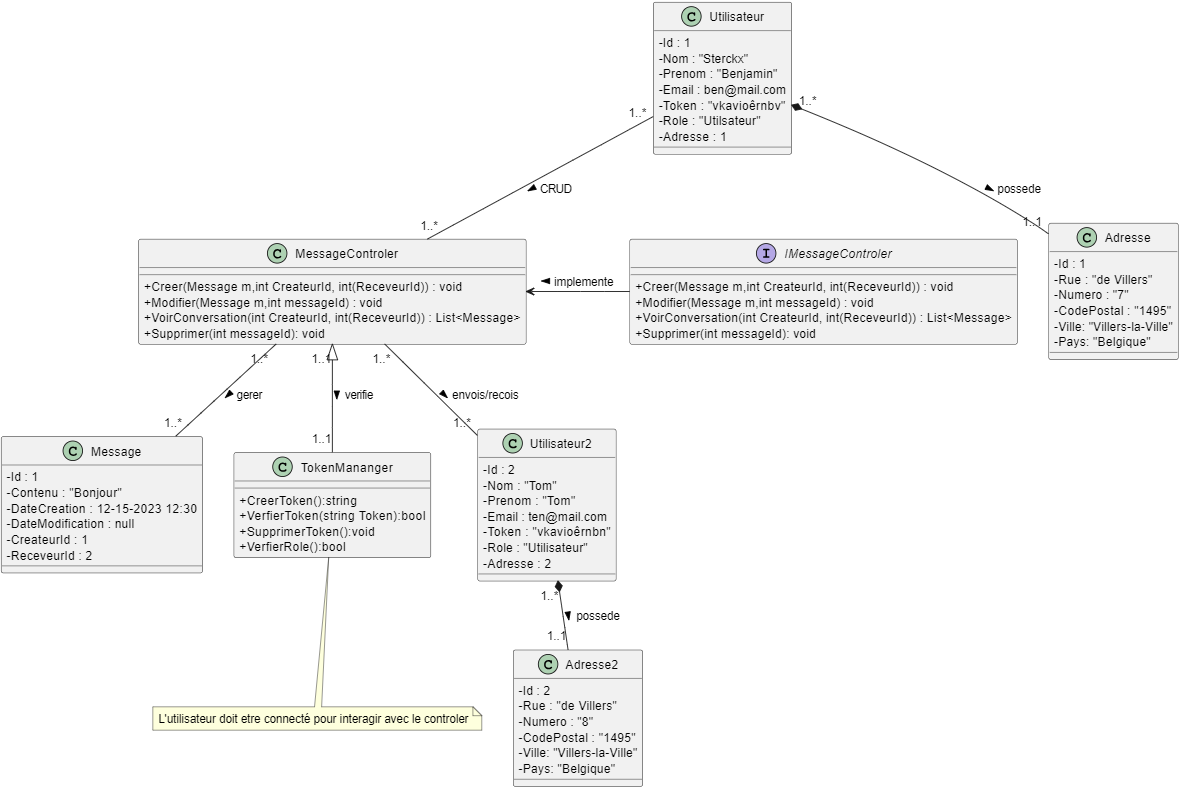
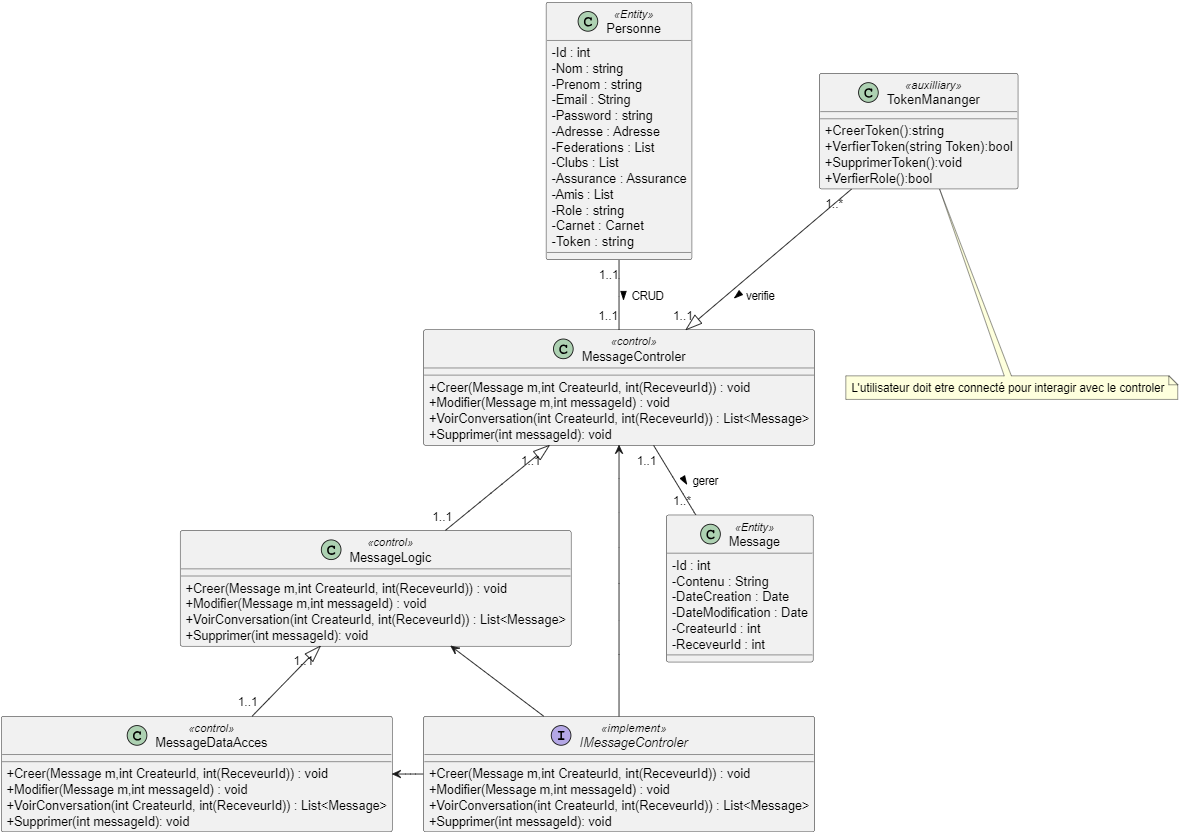


Diagramme de classes



### 4.3.5. Gérer son profil Description du scenario

L’utilisateur clic sur sa photo et est rediriger avec la page profil.

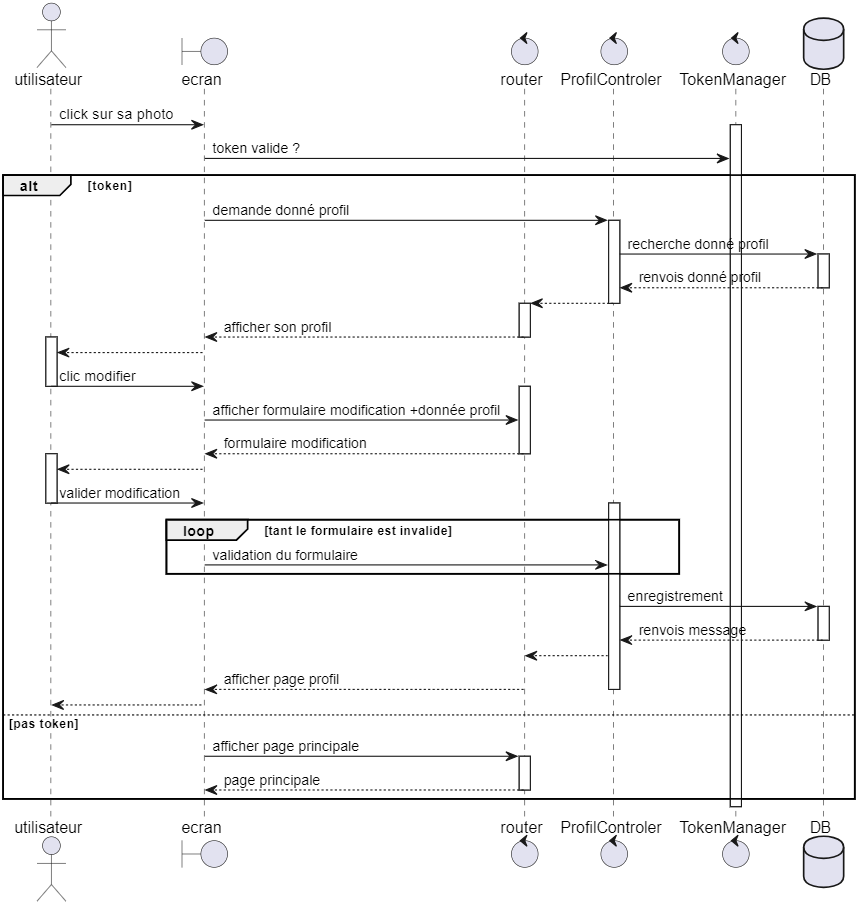
Sur cette page il a accès a toutes les infos qu’il a encodé lors de son inscription

Il peut modifier des champs et/ou ajouter les champs qui ne sont pas requit.

Il peut désactiver se déconnecter

|  |  |
| --- | --- |
| 1. L’utilisateur clic sur sa photo | 2. redirection vers la page profil |
| 3. il peut modifier/ajouter les champs | 4. L’info est mémorisé dans la base de donnée et puis il est redirigé vers son prof |

Diagramme de séquences



Scenario alternatif et Diagramme de séquences alternatif

|  |  |
| --- | --- |
| 1. L’utilisateur clic sur sa photo | 2. redirection vers la page profil |
| 3. il peut déactiver sont profil | 4. L’info est mémorisé dans la base de donnée, il ne sera plus visible par les autres utilisateurs |

### 

Diagramme de classes participatives

### 

Diagramme de classes

### 

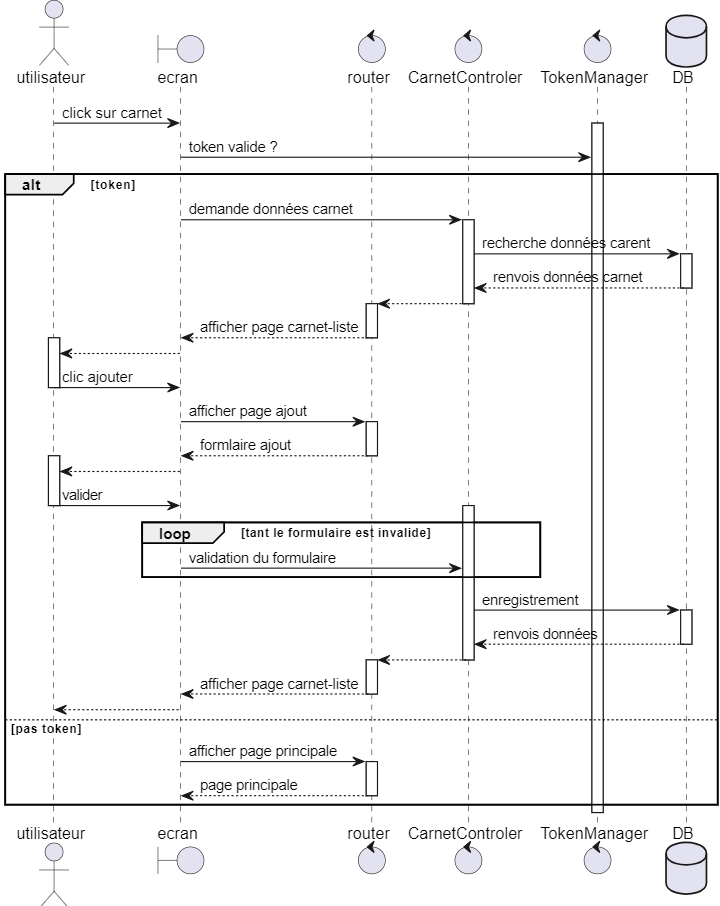
### 4.3.6. Gérer son carnet de plongée

Description du scenario

L’utilisateur clic carnet et est rediriger avec la page carnet. Sur cette page il a accès a une liste de ses plongées. Il peut ajouter un plongée a son carnet. Il peut modifier un plongée en cliquant dessus il peut supprimer une plongée de son carnet

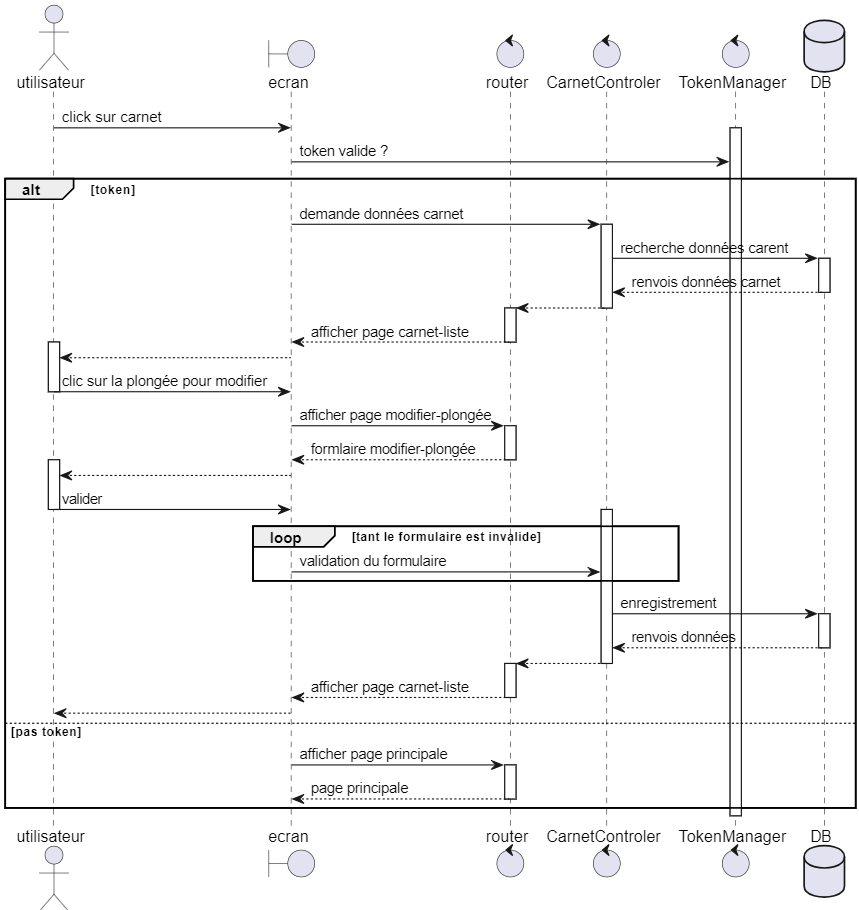
|  |  |
| --- | --- |
| 1. L’utilisateur clic sur carnet | 2. Redirection vers la page carnet |
| 3. Il peut ajouter une plongée a son carnet | 4. Il est redirigé vers la page ajoute-plongée |
| 5. Il encode sa nouvelle plongée puis valide | 6. L’info est mémorisé dans la base de donnée. Puis il est rediriger vers sa liste ou la nouvelles plongée sera visible. |

Diagramme de séquences



Scenario alternatif et Diagramme de séquences alternatif

|  |  |
| --- | --- |
| 1. L’utilisateur clic sur carnet | 2. Redirection vers la page carnet |
| 3. Il peut modifier une plongée a son carnet en cliquant dessus | 4. Il est redirigé vers la page modifier-plongée,les info cette plongées seront également redirigée |
| 5. Il modifie sa plongée puis valide | 6. L’info est mémorisé dans la base de donnée. Puis il est rediriger vers sa liste ou la plongée modifié sera visible. |



|  |  |
| --- | --- |
| 1. L’utilisateur clic sur carnet | 2. redirection vers la page carne |
| 3. Il peut supprimer une plongée de son carnet en cliquant supprimer | 4. L’info est mémorisé dans la base de donnée. Puis il est rediriger vers sa liste ou la plongée ne sera visible. |

### 

Diagramme de classes participatives

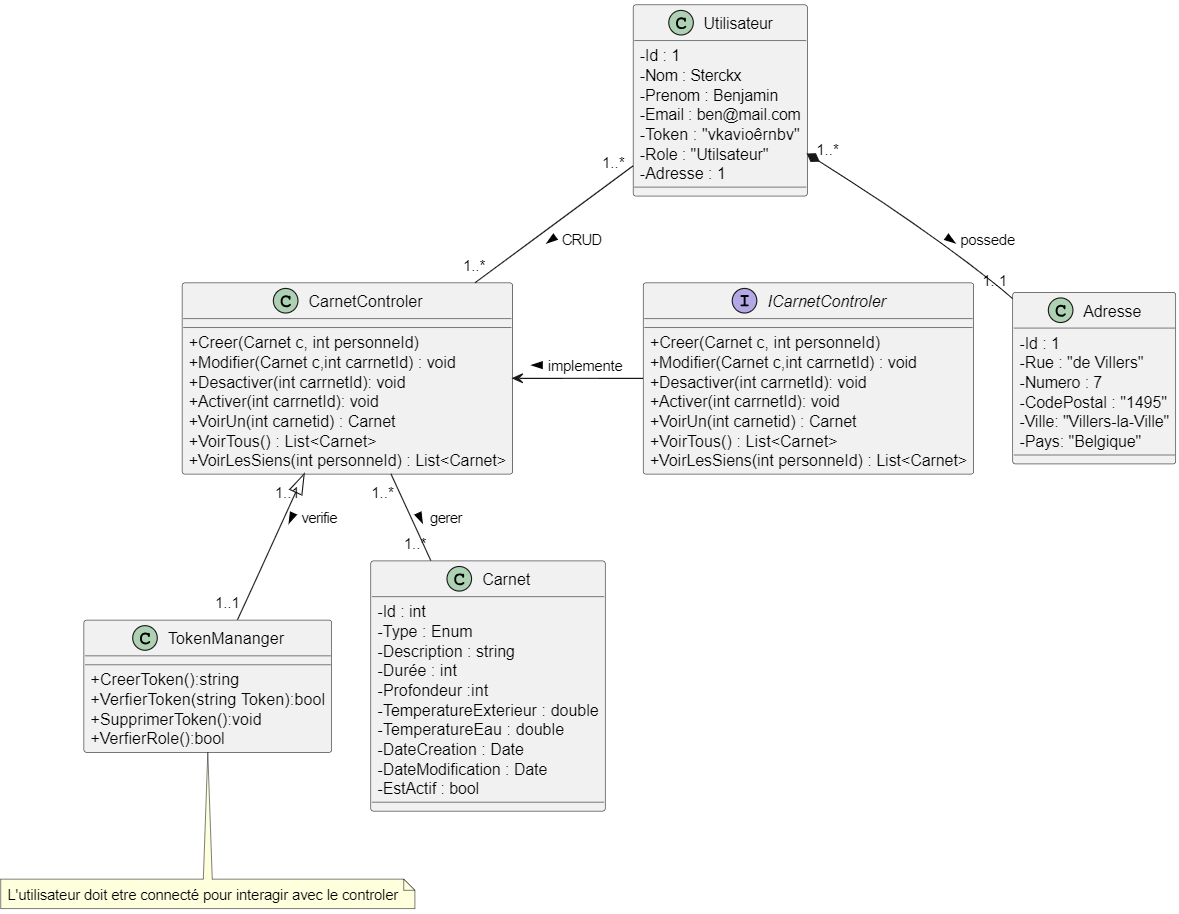
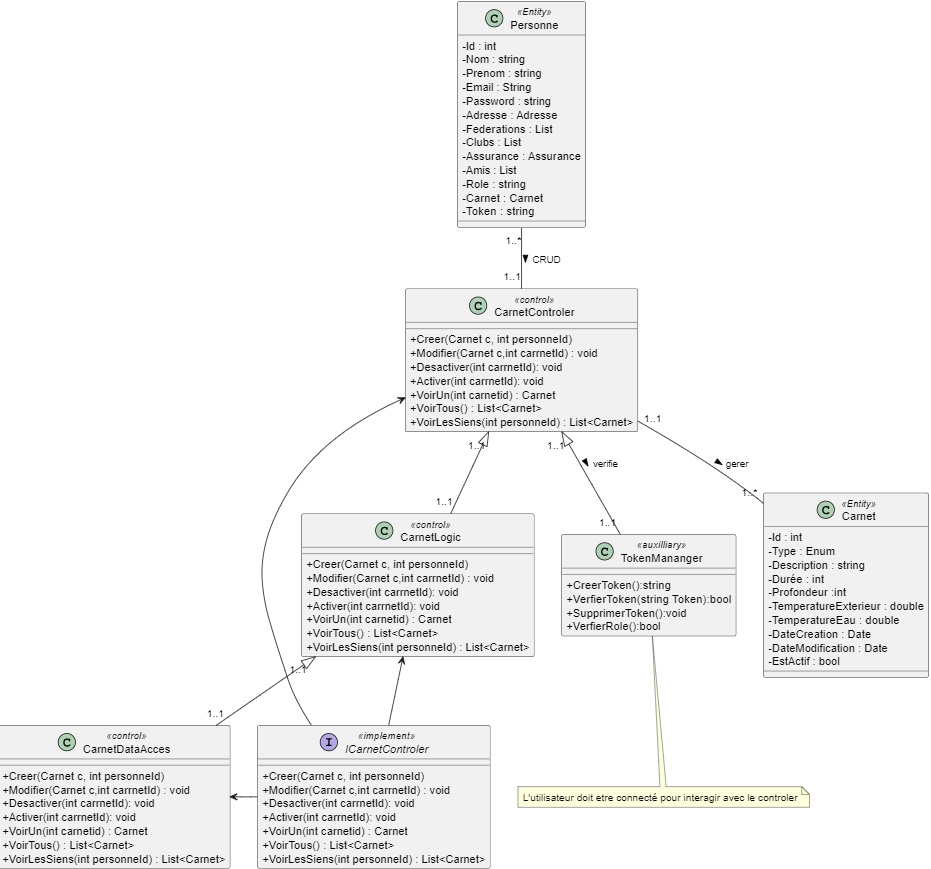


Diagramme de classes



### 4.3.7. Gérer les clubs de plongée

Description du scenario

L’admin clic club et est rediriger avec la page club.

Sur cette page il a accès a une liste de clubs.

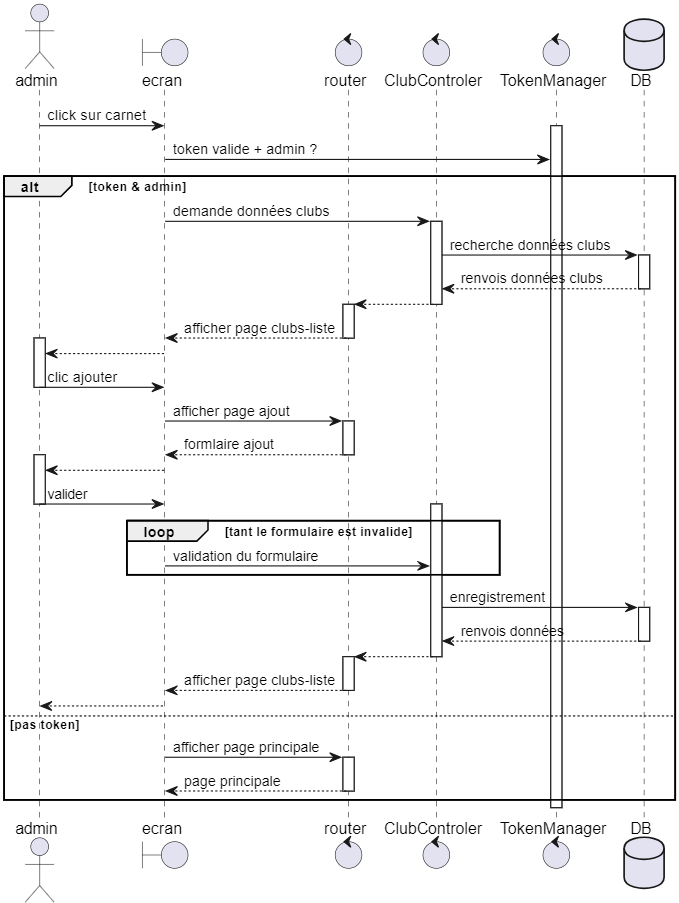
Il peut ajouter un club.

Il peut modifier un club en cliquant dessus

Il peut supprimer une club de la liste

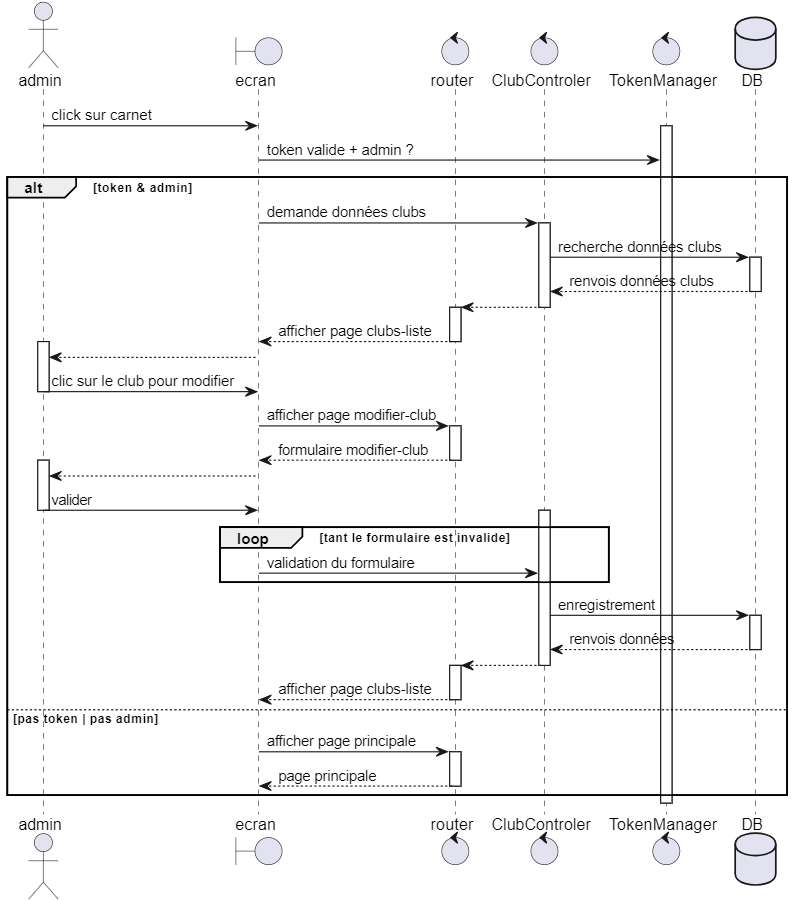
|  |  |
| --- | --- |
| 1. L’utilisateur clic sur club | 2. Redirection vers la page club |
| 3. Il peut ajouter un club | 4. Il est redirigé vers la page ajoute-club |
| 5. Il encode un nouveau club puis valide | 6. L’info est mémorisé dans la base de donnée. Puis il est rediriger vers la liste ou le club sera visible. |

Diagramme de séquences



Scenario alternatif et Diagramme de séquences alternatif

|  |  |
| --- | --- |
| 1. L’utilisateur clic sur club | 2. Redirection vers la page club |
| 3. Il peut modifier un club | 4. Il est redirigé vers la page modifier-club |
| 5. Il modifier le club puis valide | 6. L’info est mémorisé dans la base de donnée. Puis il est rediriger vers la liste ou le club sera visible |



|  |  |
| --- | --- |
| 1. L’admin clic sur club | 2. Redirection vers la page club |
| 3. Il peut supprimer un club en cliquant supprimer | 4. L’info est mémorisé dans la base de donnée. Puis il est rediriger vers la liste ou le club ne sera plus présent |

### 

Diagramme de classes participatives

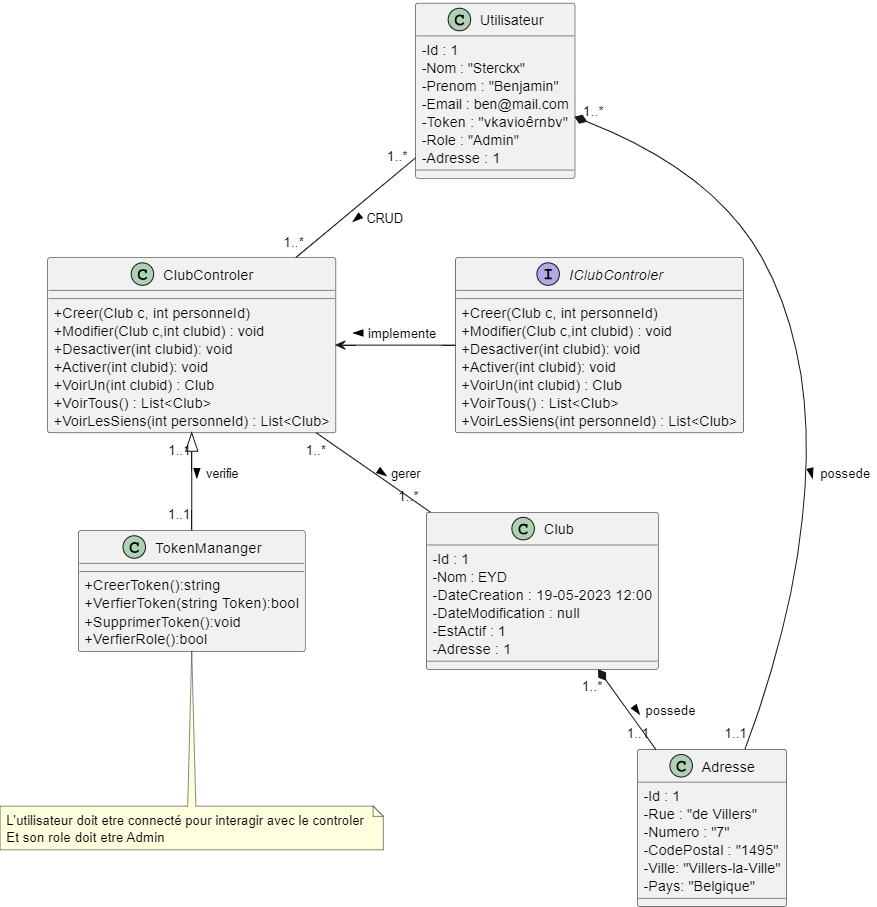
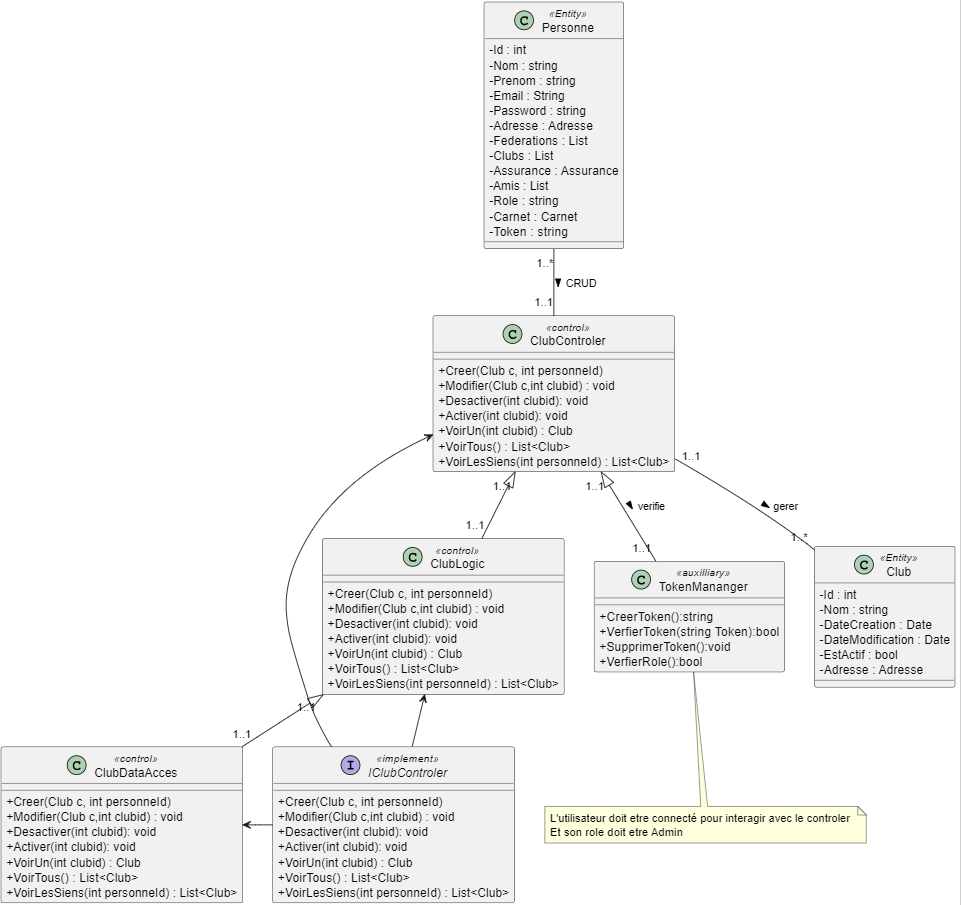


Diagramme de classes



### 4.3.8. Gérer les sites de plongée

Description du scenario

L’admin clic site et est rediriger la page sites de plongée.

Sur cette page il a accès a une liste de sites de plongée.

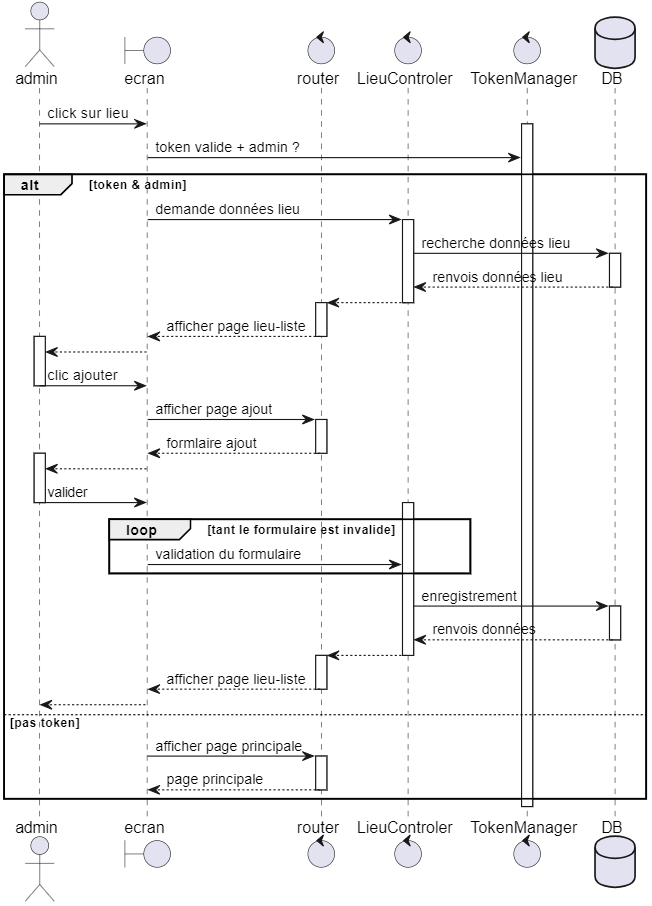
Il peut ajouter un site.

Il peut modifier un site.

Il peut supprimer une site.

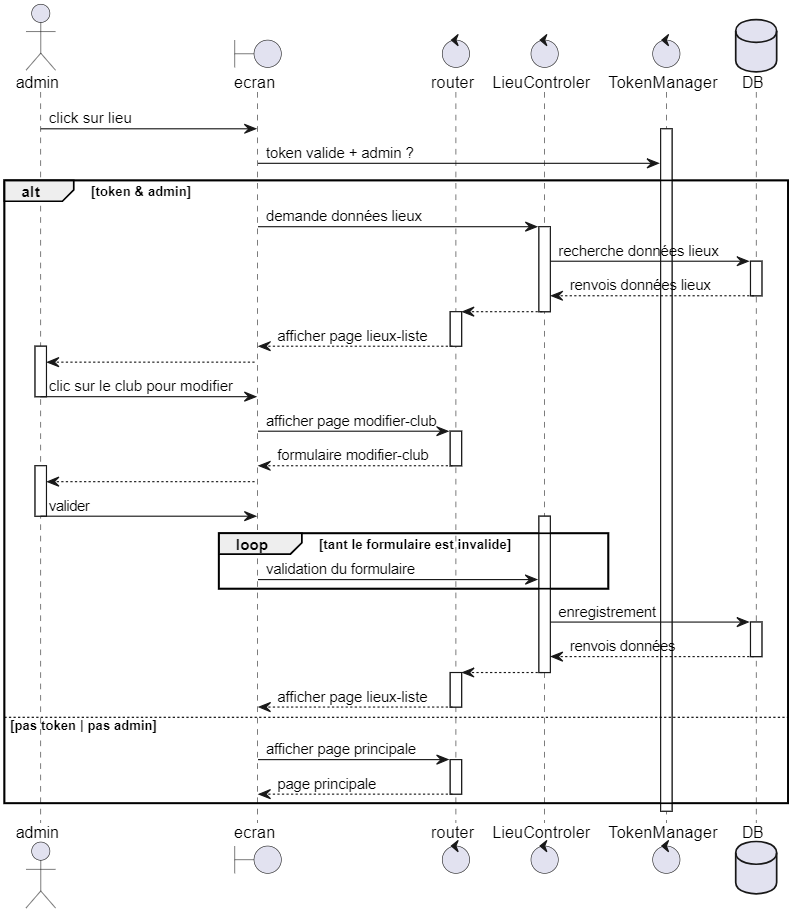
|  |  |
| --- | --- |
| 1. L’utilisateur clic sur site | 2. Redirection vers la page sites de plongée |
| 3. Il peut ajouter un site de plongée | 4. Il est redirigé vers la page insert-site |
| 5. Il encode un nouveau site puis valide | 6. L’info est mémorisé dans la base de donnée. Puis il est rediriger vers la liste ou le site sera visible. |

Diagramme de séquences



Scenario alternatif et Diagramme de séquences alternatif

|  |  |
| --- | --- |
| 1. L’admin clic sur lieu | 2. Redirection vers la page lieu |
| 3. Il peut modifier un lieu | 4. Il est redirigé vers la page modifier-lieu |
| 5. Il modifie le lieu puis valide | 6. L’info est mémorisé dans la base de donnée. Puis il est rediriger vers la liste ou le lieu sera visible |



|  |  |
| --- | --- |
| 1. L’admin clic sur lieu | 2. Redirection vers la page lieu |
| 3. Il peut supprimer un lieu en cliquant supprimer | 4. L’info est mémorisé dans la base de donnée. Puis il est rediriger vers la liste ou le lieu ne sera plus présent. |

### 

Diagramme de classes participatives

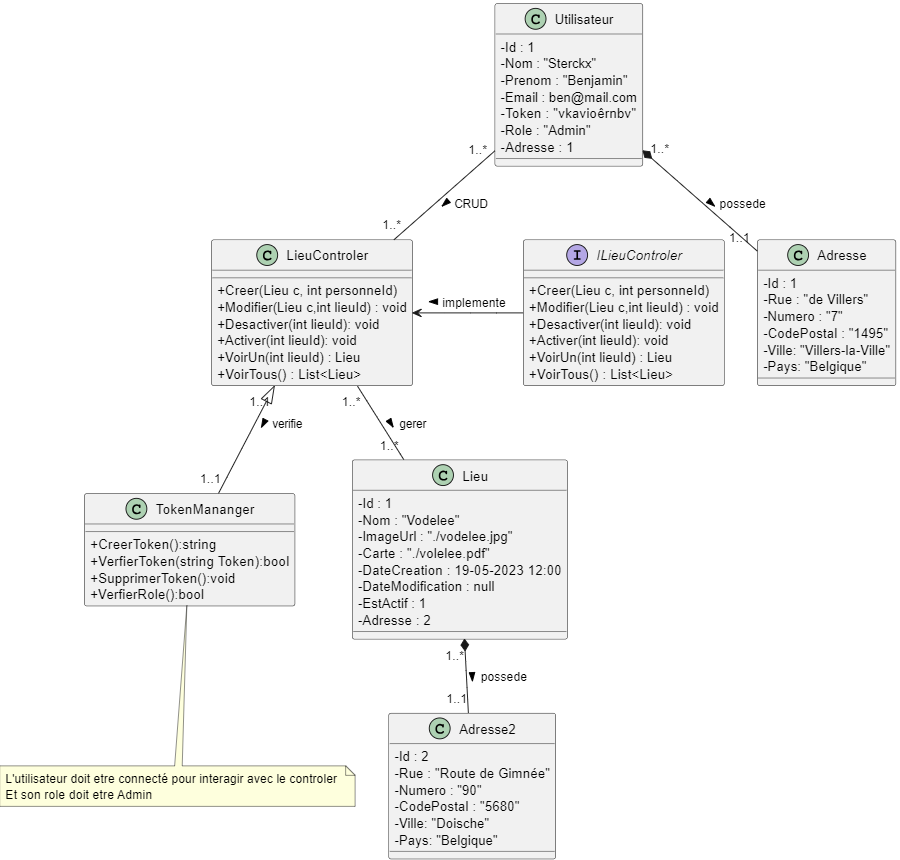
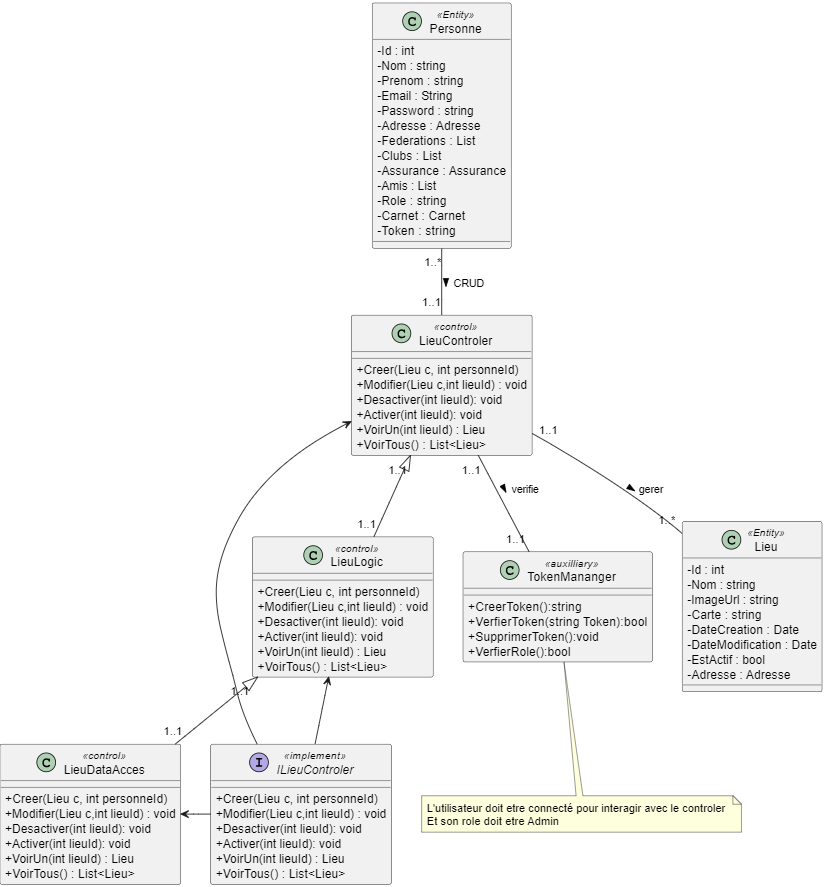


Diagramme de classes



### 4.3.9. Gérer les fédérations de plongée

Description du scenario

L’admin clic fédération et est rediriger avec la page fédération.

Sur cette page il a accès a une liste de fédérations.

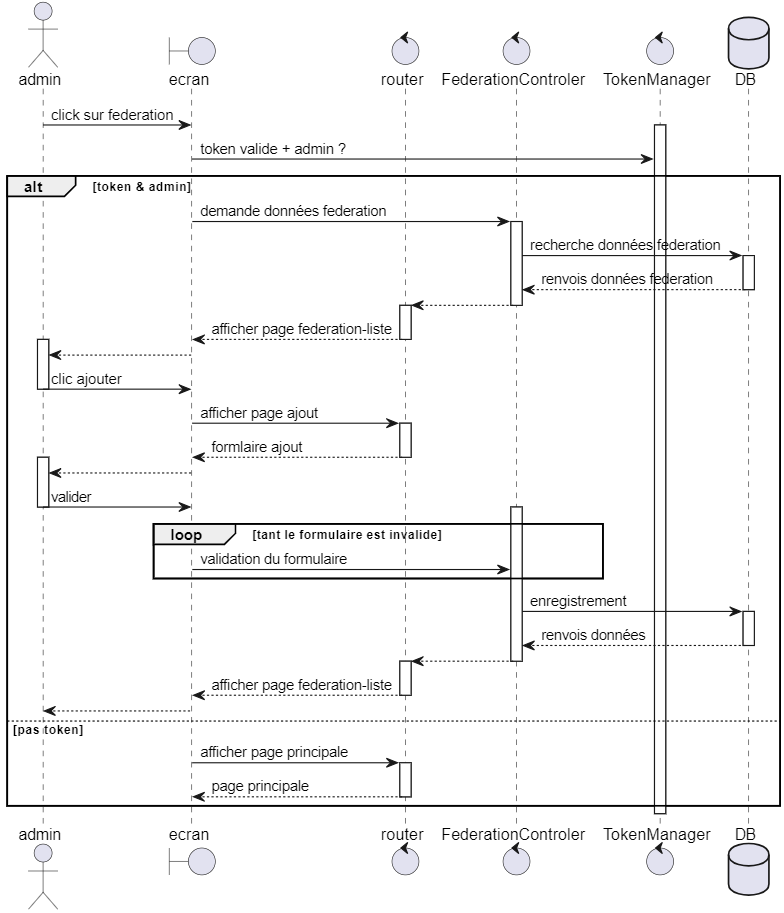
Il peut ajouter une fédération.

Il peut modifier une fédération en cliquant dessus

Il peut supprimer une fédération

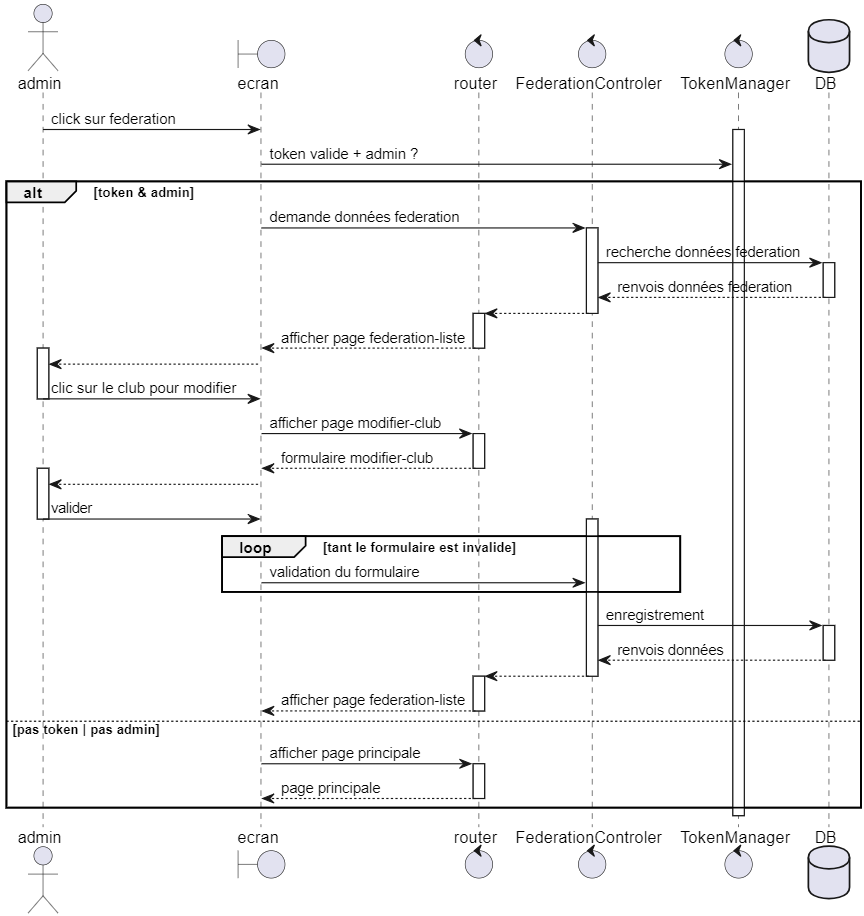
|  |  |
| --- | --- |
| 1. L’admin clic sur fédération | 2. Redirection vers la page fédération |
| 3. Il peut ajouter une fédération de plongée | 4. Il est redirigé vers la page ajoute-fédération |
| 5. Il encode une nouvelle fédération puis valide | 6. L’info est mémorisé dans la base de donnée. Puis il est rediriger vers la liste ou la fédération sera visible. |

Diagramme de séquences



Scenario alternatif et Diagramme de séquences alternatif

|  |  |
| --- | --- |
| 1. L’admin clic sur fédération | 2. Redirection vers la page fédération |
| 3. Il peut modifier une fédération | 4. Il est redirigé vers la page modif- fédération |
| 5. Il modifie la fédération puis valide | 6. L’info est mémorisé dans la base de donnée. Puis il est rediriger vers la liste ou la fédération sera visible. |



|  |  |
| --- | --- |
| 1. L’admin clic sur fédération | 2. Redirection vers la page fédération |
| 3. Il peut supprimer une fédération en cliquant supprimer | 4. L’info est mémorisé dans la base de donnée. Puis il est rediriger vers la liste ou la fédération ne sera plus présent. |

### 

Diagramme de classes participatives

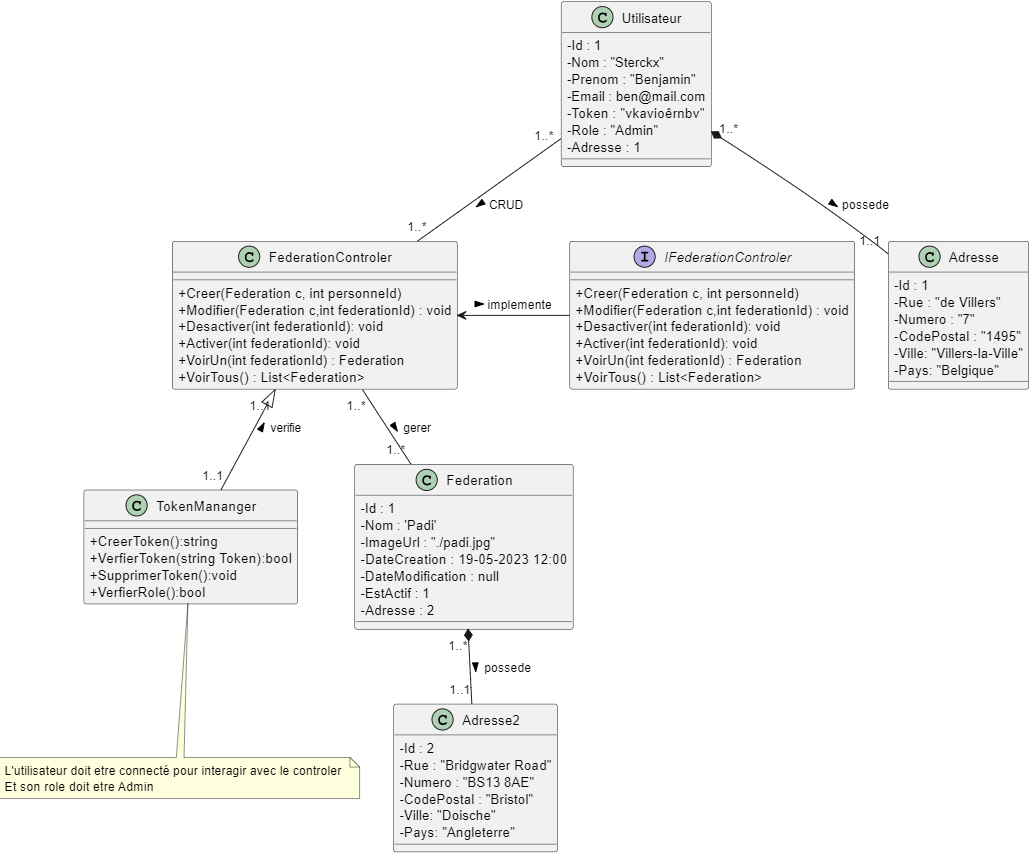
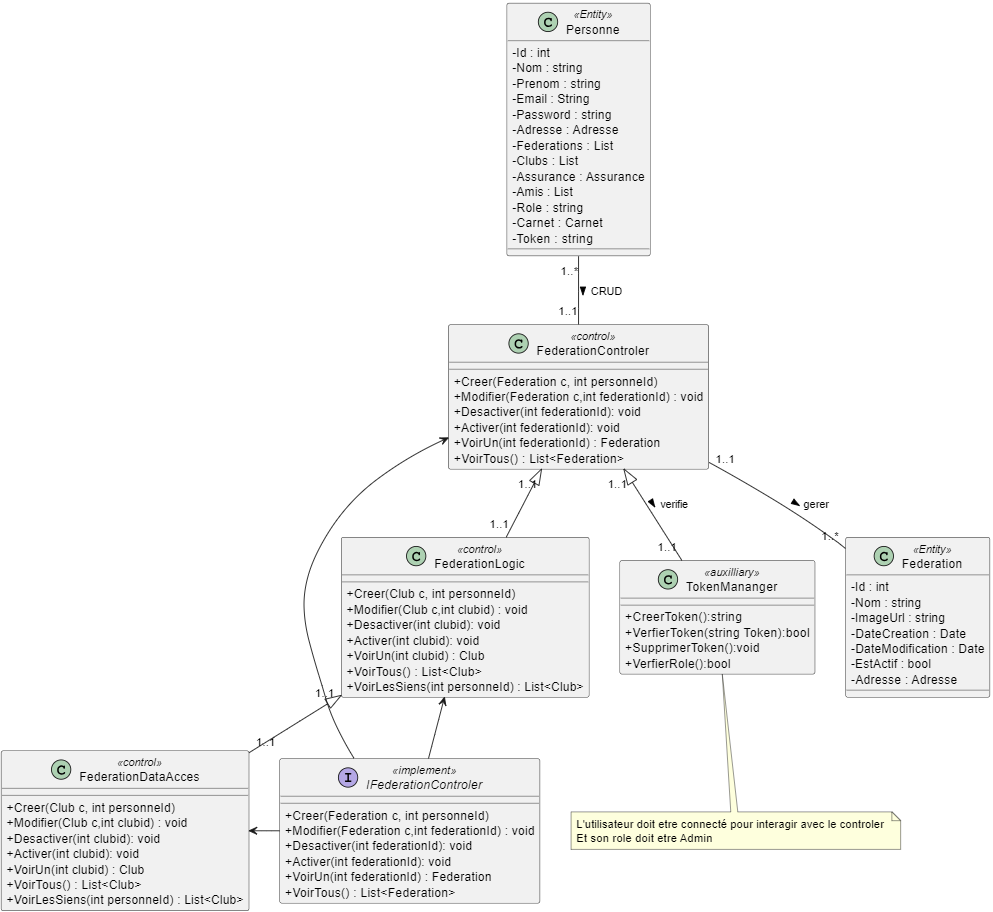


Diagramme de classes



### 4.3.10. Gérer les droit

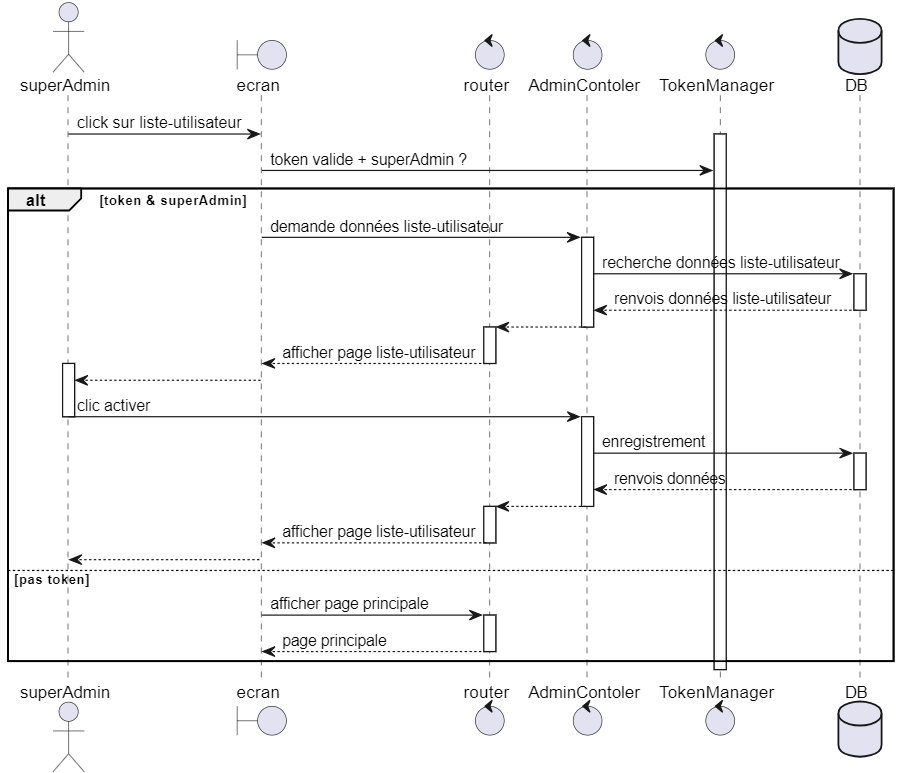
Description du scenario

Le super admin a accès a un lien liste d’utilisateur.

Sur cette page il peut attribuer des droits admin en cliquant sur activer ou désactiver

|  |  |
| --- | --- |
| 1. Le super admin clic sur liste-utilisateur | 2. Redirection vers la page liste-utilisateur |
| 3. Il peut donner les droits admin en cliquant sur activer | 4. L’info est mémorisé dans la base de donnée. Puis il est rediriger vers la liste-utilisateur et verra le statuts modifier |

Diagramme de séquences



Scenario alternatif et Diagramme de séquences alternatif

|  |  |
| --- | --- |
| 1. Le super admin clic sur liste-utilisateur | 2. redirection vers la page liste-utilisateur |
| 3. Il peut enlever les droits admin en cliquant sur déactiver | 4. L’info est mémorisé dans la base de donnée. Puis il est rediriger vers la liste-utilisateur et verra le statuts modifier |

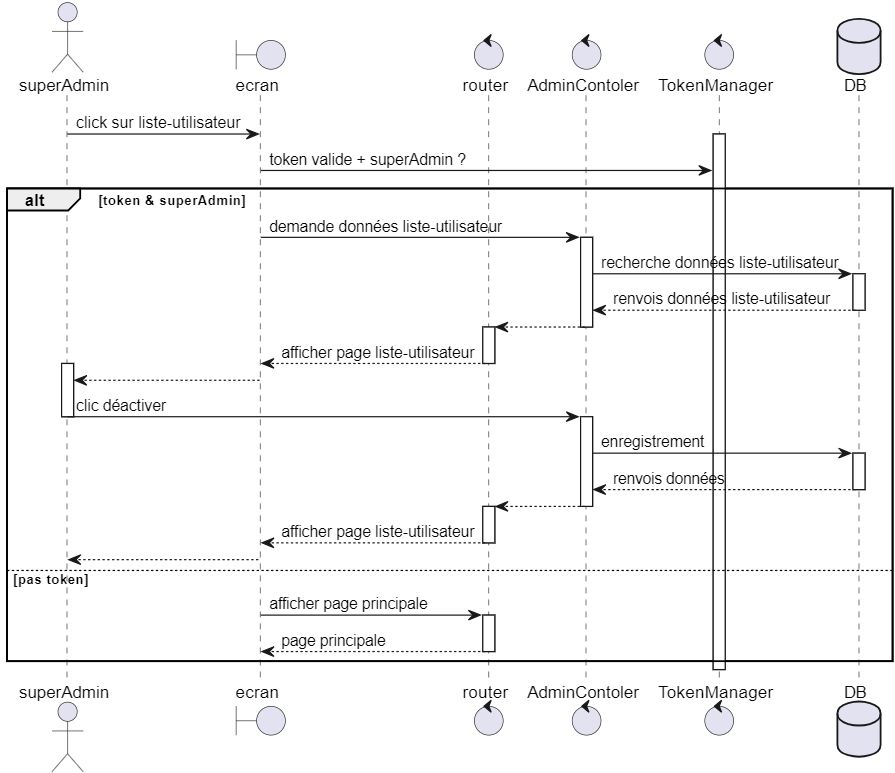


Diagramme de classes participatives

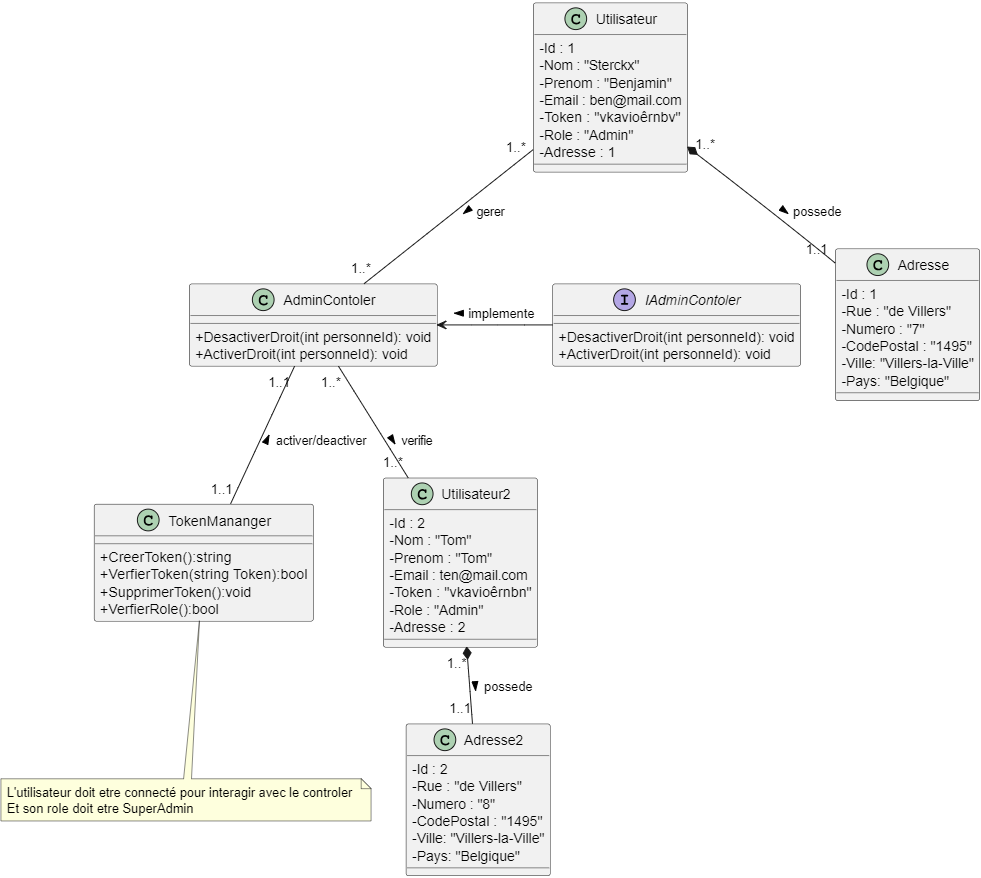
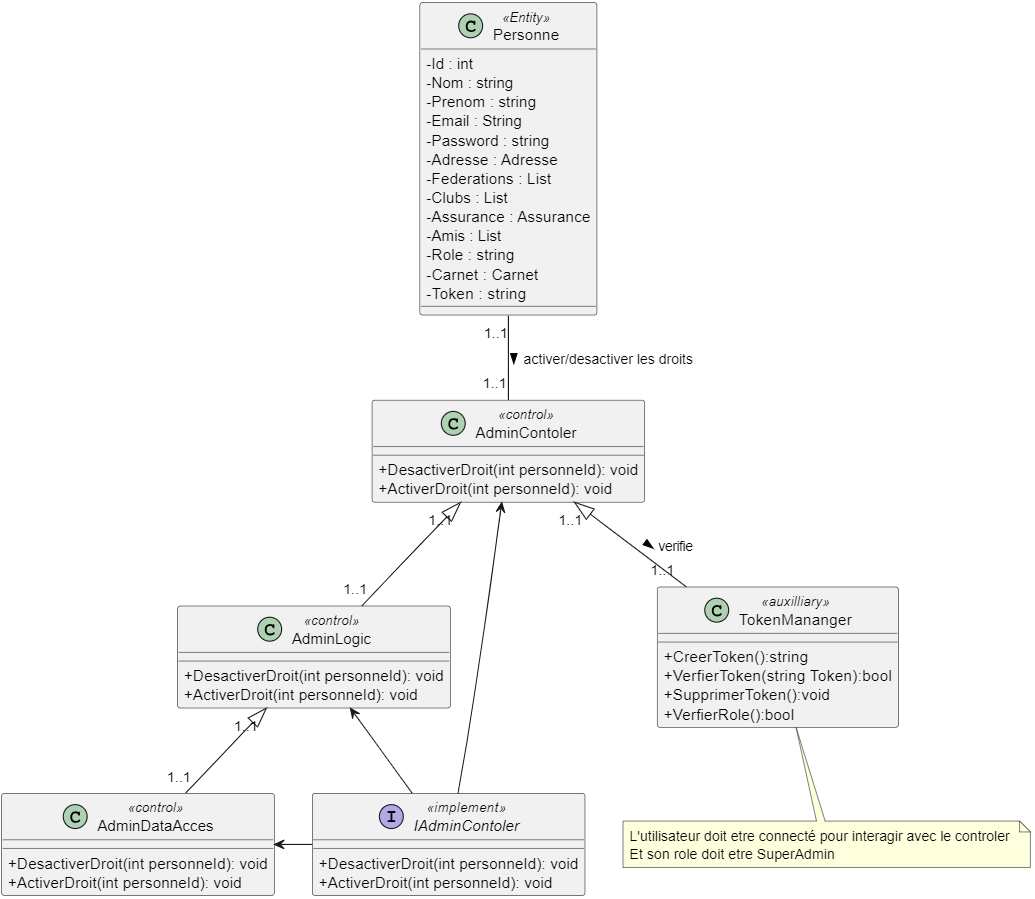


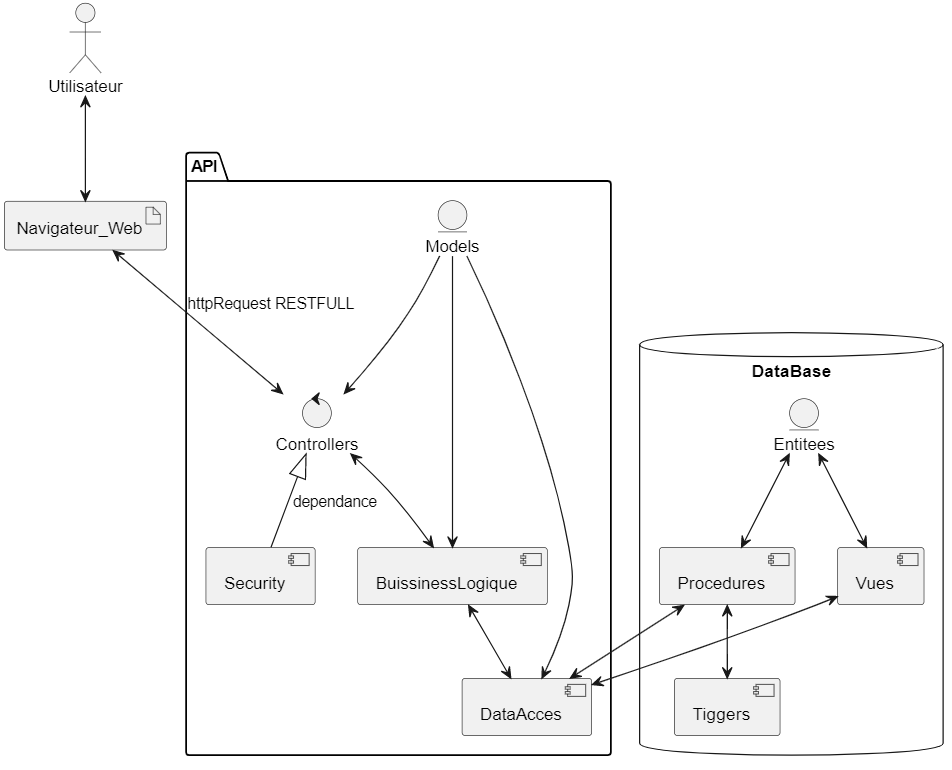
Diagramme de classes



## 4.4. Diagramme de packages



## 4.5. Diagramme de déploiement



## 4.6. Choix des outils informatiques

### 4.6.1. La Base de données

Pour la partie base de données, j’ai utilisé SQL Server

Microsoft SQL Server est un système de gestion de base de données (SGBD) en langage SQL incorporant entre autres un SGBDR (SGBD relationnel ») développé et commercialisé par la société Microsoft. Il fonctionne sous les OS Windows et Linux depuis mars 2016.

https://fr.wikipedia.org/wiki/Microsoft\_SQL\_Server

Source d’information pour SQL Server : <https://learn.microsoft.com/fr-fr/sql/relational-databases/tutorial-getting-started-with-the-database-engine?view=sql-server-ver16>

### 4.6.2. Le Server

Pour la partie server, j’ai utilisé le framework .NET6  
.NET (prononcé « dot net », anciennement .NET Core) est un framework libre et open source développé par les employés de Microsoft regroupés dans la .NET Foundation (en).Il est compatible avec les systèmes d'exploitation Windows, macOS et Linux

https://fr.wikipedia.org/wiki/.NET

Source d’information sur .NET : <https://learn.microsoft.com/fr-fr/dotnet/>

### 4.6.3. Le Client

Pour la partie client, j’ai utilisé le framework Angular pour développer l'interface utilisateur de l'application.

Angular est un framework d'application Web d'une seule page basé sur TypeScript , gratuit et open source, exécuté sur Node.js. Il est dirigé par l'équipe Angular de Google et par une communauté d'individus et d'entreprises. L'écosystème angulaire se compose d'un groupe diversifié de plus de 1,7 million de développeurs, auteurs de bibliothèques et créateurs de contenu.

https://en.wikipedia.org/wiki/Angular\_(web\_framework)

Source d’information pour Angular : <https://angular.io/>

### 4.6.4. Les interfaces de développement

Visual Studio Code

Coté client, j’ai utilisé Visual Studio Code pour codé Angular.  
Grace a son outil extension, j’ai pu installer quelques logiciels bien utile comme :

- Material Icon Theme

Permet d’avoir un icon qui represente l’extension du fichier un l’arborescence.

- Angular Schematics

Permet de créer des éléments Angular par interface graphique

- Angular Snippets

Auto complétion du code orienté Angular

- Auto Rename Tag

Lors d’un modification de balise cela modifier l’ouverture comme la fermeture

- Azure App Service

Permet de me connecter a mon compte Azure afin de deployer l’application

- French Language Pack for Visual Studio Code:

Permet d’avoir l’interface Visual Studio Code en francais

- Plant Uml:

permet de créer des shema uml grace au code Plant uml

Ensuite par ligne de commande j’ai pu installer les dépendances suivant :

- Bootstrap version 5.2.3

Permet d’accéder a une librairie CSS

Informations : https://getbootstrap.com/docs/5.0/getting-started/introduction/

- Bootstrap-icons version 1.11.3

Permet d’accéder a une librairie d’icônes

Informations : <https://icons.getbootstrap.com/>

- rxjs version 7.5.0

Bibliothèque de programmation réactive utilisant des observables

Informations : <https://rxjs.dev/>

Visual Studio

Coté server, j’ai utiliser Visual Studio pour codé .NET6.

Grace au gestionnaire de package NuGet, j’ai pu installer quelques dépendances bien utile comme :

- Bcrypt-Official

Un port .Net de jBCrypt implémenté en C#. Il utilise une variante du programme de chiffrement de l'algorithme de chiffrement Blowfish et introduit un facteur de travail qui vous permet de déterminer le coût de la fonction de hachage, permettant ainsi à l'algorithme d'être « évolutif ».

Informations : <https://www.nuget.org/packages/BCrypt-Official>

- Microsoft.AspNetCore.Authentication.JwBearer

Middleware ASP.NET Core qui permet à une application de recevoir un jeton de porteur OpenID Connect.

Informations : https://www.nuget.org/packages/Microsoft.AspNetCore.Authentication.JwtBearer

- Microsoft.Extensions.Logging.Abstractions

Ce package comprend un générateur de source de journalisation qui produit un code hautement efficace et optimisé pour les méthodes de journalisation des messages.

Informations : https://www.nuget.org/packages/Microsoft.Extensions.Logging.Abstractions/

- Microsoft.IdentityModel.Tokens

Inclut les types qui prennent en charge les SecurityTokens et les opérations cryptographiques : signature, vérification des signatures, cryptage.

Informations : https://www.nuget.org/packages/Microsoft.IdentityModel.Tokens/

- System.Data.SqlClient

Fournit le fournisseur de données pour SQL Server. Ces classes donnent accès aux versions de SQL Server et encapsulent des protocoles spécifiques à la base de données, y compris le flux de données tabulaires (TDS).

Informations : https://www.nuget.org/packages/System.Data.SqlClient

- System.Net.Http

Fournit une interface de programmation pour les applications HTTP modernes, y compris des composants client HTTP qui permettent aux applications de consommer des services Web via HTTP et des composants HTTP qui peuvent être utilisés à la fois par les clients et les serveurs pour analyser les en-têtes HTTP.

Informations : https://www.nuget.org/packages/System.Net.Http/

SQL Server Management Studio

Coté base de données, j’ai utiliser SQL Server Management Studio.

Microsoft SQL Server Management Studio est une application logicielle développée par Microsoft qui est utilisée pour configurer, gérer et administrer tous les composants de Microsoft SQL Server

Informations:

https://learn.microsoft.com/en-us/sql/ssms/quickstarts/ssms-connect-query-sql-server?view=sql-server-ver16

Github

Afin de sauvegarder mon travail , j’ai utilisé Github

GitHub est une plateforme de développement qui permet aux développeurs de créer, stocker, gérer et partager leur code. Il utilise le logiciel Git , fournissant le contrôle de version distribué de Git ainsi que le contrôle d'accès , le suivi des bogues , les demandes de fonctionnalités logicielles , la gestion des tâches , l'intégration continue et les wikis pour chaque projet.

Informations: https://github.com/about

## **4.7. Méthodoligie de travail utilisée**

### 4.7.1 Mérise

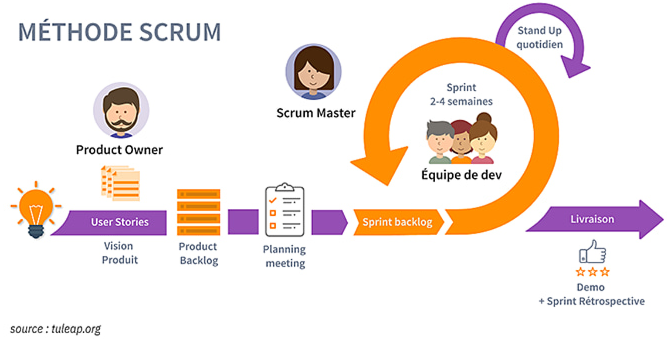
La méthode Merise est une méthode d'analyse, de conception et de réalisation de systèmes d'informations. Au niveau conceptuel, elle vise à décrire, après abstraction, le modèle (le système) de l'entreprise ou de l'organisme afin de produire un schéma permettant de construire notre base de données.

https://fr.wikipedia.org/wiki/Merise\_(informatique)

### 4.7.2 Scrum

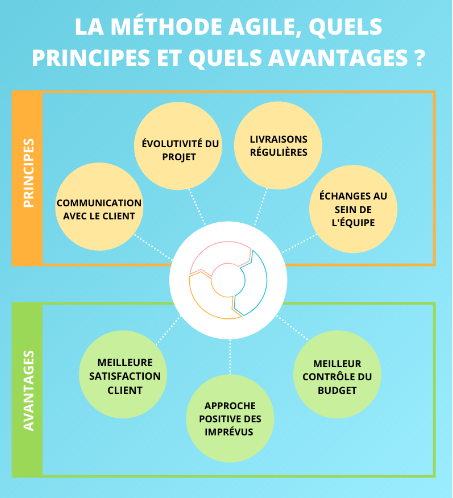
Scrum est considéré comme un groupe de pratiques répondant pour la plupart aux préconisations du manifeste agile.  
Scrum s'appuie sur le découpage d'un projet en « boîtes de temps », nommées sprints (« pointes de vitesse »). Les sprints peuvent durer entre quelques heures et un mois (avec un sprint médian à deux semaines). Chaque sprint commence par une estimation suivie d'une planification opérationnelle. Le sprint se termine par une démonstration de ce qui a été achevé.

https://fr.wikipedia.org/wiki/Scrum\_(d%C3%A9veloppement)



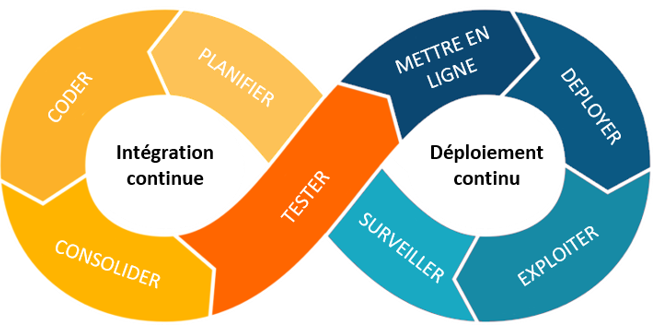
### 4.7.3 Agile

En ingénierie logicielle, les pratiques agiles mettent en avant la collaboration entre des équipes auto-organisées et pluridisciplinaires et leurs clients. Elles s'appuient sur l'utilisation d'un cadre méthodologique léger mais suffisamment centré sur l'humain et la communication. Elles préconisent une planification adaptative, un développement évolutif, une livraison précoce et une amélioration continue, et elles encouragent des réponses flexibles au changement.

1. https://fr.wikipedia.org/wiki/M%C3%A9thode\_agile
2. ****

### 4.7.4 CI/CD

[Le CI/CD comble le fossé entre les activités et les équipes de développement et d'exploitation en imposant l'automatisation de la création, des tests et du déploiement des applications. Les pratiques DevOps modernes impliquent le développement continu, le test continu, l'intégration continue, le déploiement continu et la surveillance continue des applications logicielles tout au long de leur cycle de vie. La pratique CI/CD, ou pipeline CI/CD, constitue l'épine dorsale des opérations DevOps modernes.](https://fr.wikipedia.org/wiki/Méthode_agile)

1. https://fr.wikipedia.org/wiki/CI/CD
2. ****

# 5. Réalisation

* 1. Le projet a débuter par la cherche d’idées basé sur des questions simple  
     Comment répondre aux besoin de création de cette application?   
     Quels seront les outils nécessaire à cette fin?
  2. Il m’a fallut déterminer les outils qui allait être mit en place.   
     L’application étant en SAAS, Software as a Service, mon choix c’est rapidement orienté vers des outils dont j’avais une certaines maîtrise de part mes projet précédents.  
     J’ai donc choisis d’utiliser une base de données SQL Server, un Server en .NET6 et un client en Angular
  3. Dés lors, j’ai pu orienté mes recherches d’informations sur ces produits, framework  
     Ensuite est venue l’heure de l’analyse. Pour ce faire, j’ai d’abords interviewes 4 personnes afin de déterminer les besoins. J’ai utilisé Merise et Uml afin déterminer les schémas nécessaire à l’architecture du projet ceux-ci sont décrit dans le point 4.

## 5.1. L’application

### 5.1.1 L’architecture

La mise en place d’une arborescence de fichier et une nomination cohérente de ceux-ci que ce soit en en base de données, sur le sevrer ou dans la partie client m’a permis d’avoir architecture simple, compréhensible ou la recherche de fichier se fait aisément.

### 5.1.2 L'installation

SQL Server Management Studio  
[https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16#download-ssms](https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16" \l "download-ssms)

Visual Studio

<https://visualstudio.microsoft.com/fr/downloads/>

Node Js

[Node.js — Run JavaScript Everywhere (nodejs.org)](https://nodejs.org/en/)

Angular par ligne de commande

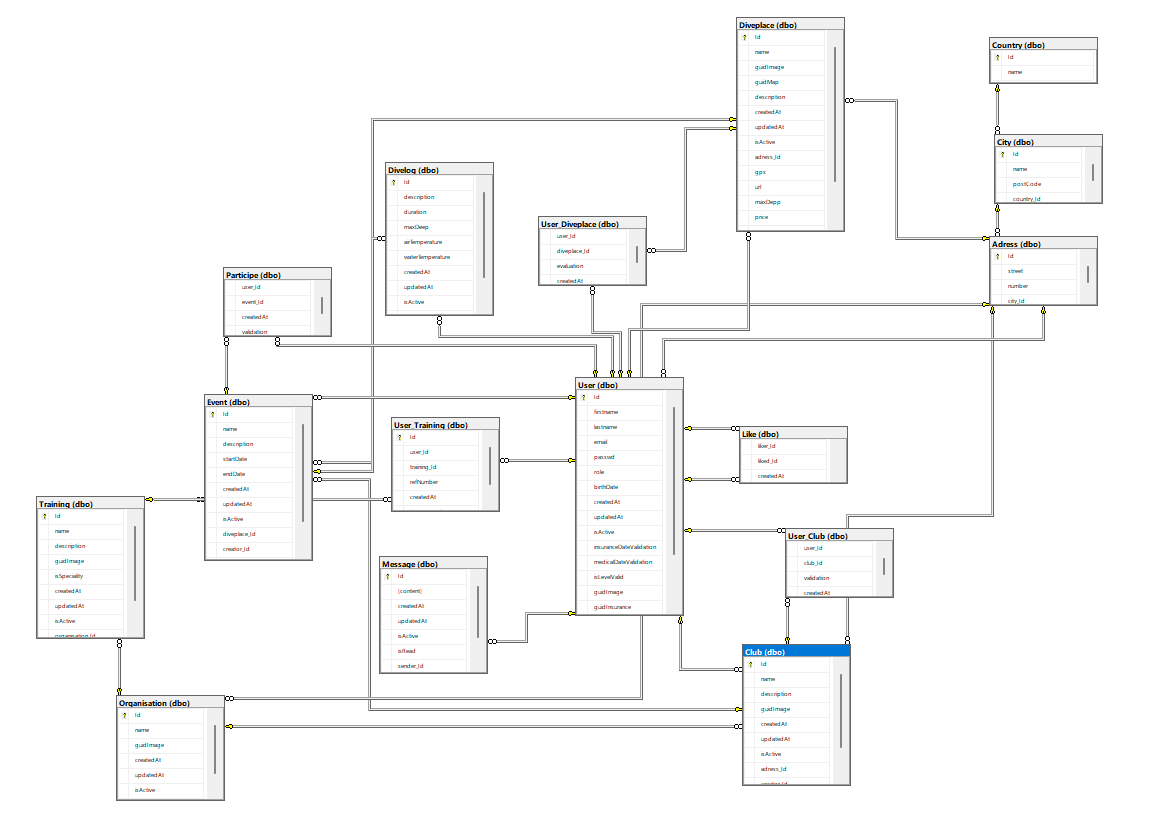
npm install -g @angular/cli

Visual Studio Code

[Visual Studio Code - Code Editing. Redefined](https://code.visualstudio.com/)

### 5.1.3 La base de donnée

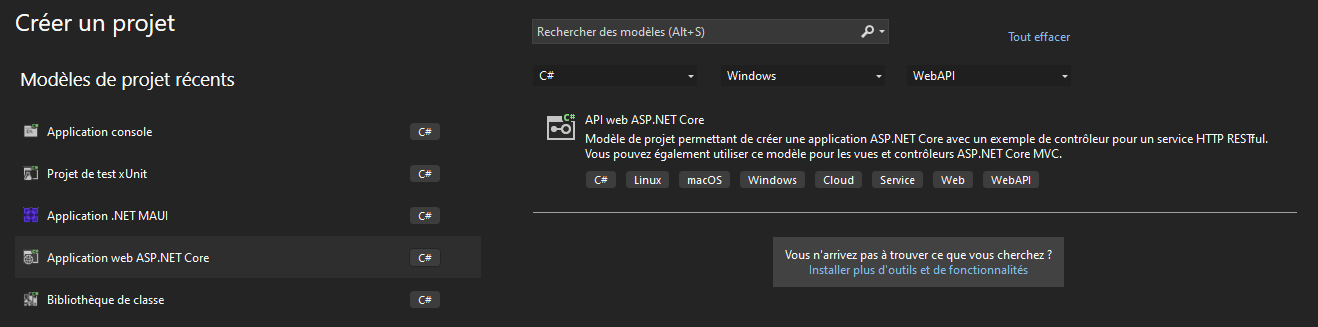
Grace à la méthode Mérise décrite au point 4 et le modèle conceptuel de données créé, j’ai pu écrire un script de création afin de créer les tables et leurs lien par clés étrangères.



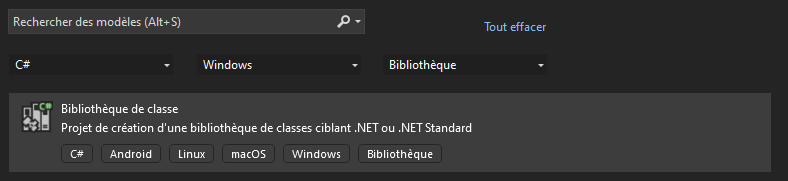
### 5.1.3 Le server

Les outils du framework .NET6 m’ont permis de créer une application web ASP.NET core facilement en suivant les informations de l’IDE lors de la création de projet.  
J’ai donc créé ce projet que j’ai nommé ‘API’. Celui-ci comportera les contrôleurs HTTP RESTfull

Les contrôleurs ont également un rôle de sécurité en effet ceux-ci n’autorisent l’accès aux fonctions que dans le cas d’utilisation d’un Token valide grâce à la configuration de fichier Programme.cs sauf bien sur pour le fonctions login et register



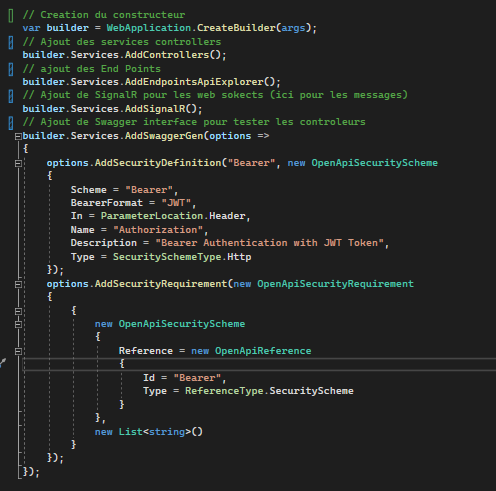
Par la suite, j’ai créer 2 autres projets de bibliothèque de classe



L’un appelé BLL pour Buisness Logique Layer, sont rôle est de créer les structure d’objet à renvoyer lors d’une requête HTTP.  
L’autre appelé DAL pour Data Access Layer, sont rôle est de communiquer et sécuriser les échanges avec la base de données.

Le projet API

C’est ‘l’entrée’ du server il détient de le fichier de configuration, le Program.cs

Pour celui-ci j’ai configurer plusieurs option

### 

### 

### 

### 5.1.5 Le client

* 1. Création du projet grâce la commande : ng new “nom\_du\_projet”
  2. Ensuite la commande ‘ng serve’ compile et démarre l’application sur le port 4200 (par défaut).

Les différents dossiers et fichiers

node\_modules: contient tout les dépendances nécessaire au fonctionnement du framework, ainsi que les

dépendances ajouté par la suite

src: dossier principal de développement

angular.json: configuration du projet

package.json: liste et version des packages utilisés dans le projet

tsconfig.json: configuration de la compilation Typescript

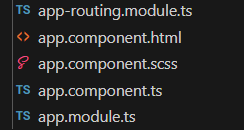
Arborescence du projet

app: on y intégrera nos composants, service, ect..

assets: contient les médias supplémentaires images, sons, vidéos, … nécessaires à l’application  
environments: permet de localiser les variables d’environnement tel que les adresses d’ API  
index.html: point d’entrée de l’application.

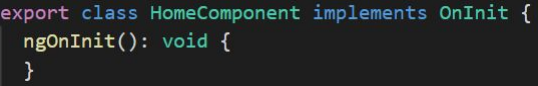
styles.scss: feuille de style par défaut de l’application

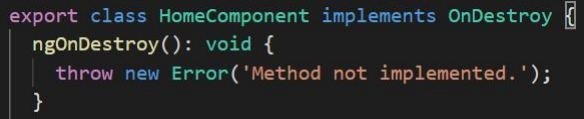
main.ts: définit le module de démarrage de l’application

Les fichiers essentiels

Les fichier app.component.html, app.component.scss et app.component.ts représentent le composant principal de la single page application

exemple de component.ts

  
  
  
  
  
  
  
ici le HomeComponent implémente ngOnInit()  
OnInit => Exécute le contenu de la méthode ngOnInit() à l’initialisation du composant



ici le HomeComponent implémente ngOnDestroy()

OnDestroy => Exécute le contenu de la méthode ngOnDestroy() à la destruction du composant  
  
Ce sont les 2 fonctions que j’ai utilisé dans chaque composant créer

Le fichier app.module.ts représente la configuration du module principal



Le décorateur @NgModule signifie au système que le fichier est le point d’entrée du module

declarations : [] => Tableau des composants utilisé(s) par le module

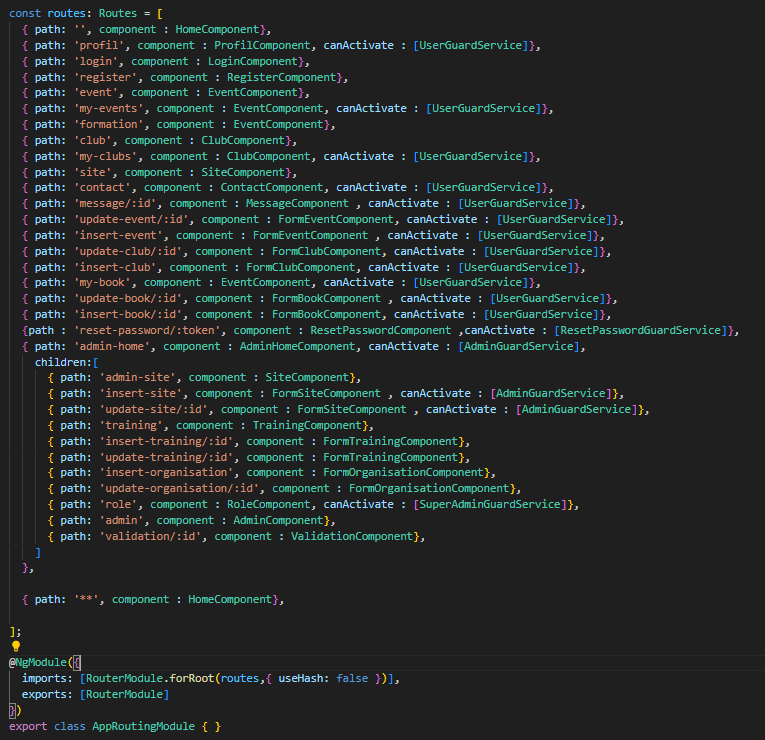
imports : [] => Tableau des différents modules importés et utilisés.

exports : [] => Tableau des exportations de modules/composants utilisé

providers : [] => Tableau des informations pour l’injection de dépendances

bootstrap : [] => Le composant d’amorçage du module (si composant il y a)

Le fichier app-routing.module.ts représente le routeur principal. C’est dans celui-ci qui nous définissons la navigation de l’application



Commande pour créer un nouveau composant  
Se placer avec la console dans l’arborescence puis, ng generate component ‘nom\_du\_composant’

Cela génère automatiquement les fichiers .html, .css , .ts du composant et ses imports

## 5.2. Mise en service

Afin de déployer l'application et quelle soit accessible par un navigateur, j'ai choisis de la déployer sur Azure  
Cette solution a été privilégier étant donné le fait que j’utilise des outils Microsoft pour le développement tel SQL Server ou .NET6

### 5.2.1 Azure

Grace a mon compte Github, j’ai pu facilement me créer un compte sur Azure, les 2 faisant parties des services Microsoft.  
C’est en suivant ce tutoriel que j’ai créer mon premier hébergement d’application   
<https://www.youtube.com/watch?v=u_CRppLcC9k>

Toute fois, j’ai modifié par rapport à ce tutoriel légèrement la configuration à savoir   
- le lieux de l’hébergement en Europe plutôt qu’au États Unis

- l’activation de l’authentification de base SCM et FTP afin de publier via les IDE

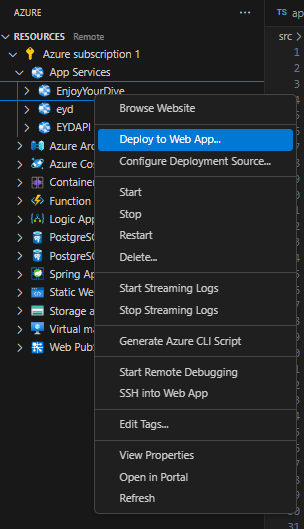
- l’activation des web sockets afin d’utiliser SingalR pour la messagerie instantanée

### 5.2.2 Les builds

Pour de déployer le code il m’a fallu ‘builder’ les applications pour la production   
Angular

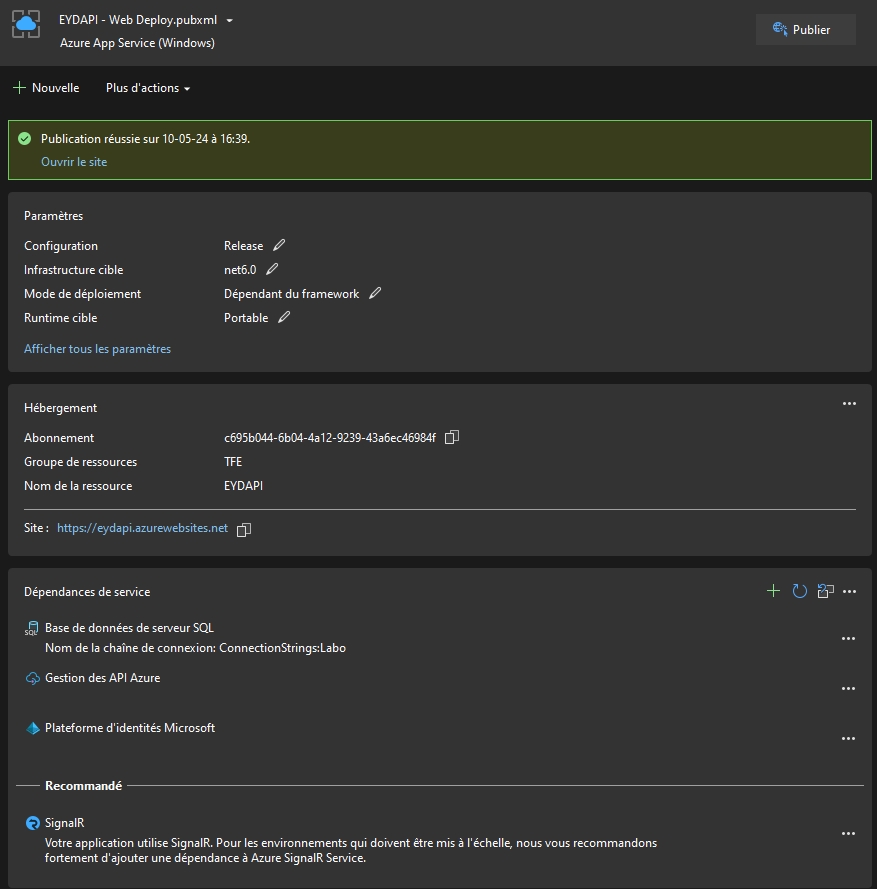
Avec la commande ng build –configuration=production

Ensuite grâce à l’extension Azure App Service, je peux via Visual Studio me connecter à Azure et déployer



ASP .NET 6

Avec les outils d’IDE : clic droit sur le projet ASP .NET et choisir publier ensuite créer un plan d’action afin de définir les liens entre base de données, server et Azure



## 5.3. Problèmes rencontrés et des solutions trouvées

### 5.3.1 Le rafraichissement du client

1. Angular étant une single page application lors d’un refresh (F5) ou d’une recompilation du code les données stocker dans la RAM sont perdues

Il me fallait systématiquement me reconnecter pour continuer à travailler  
La solution a été de créer une fonction dans le fichier app.components.ts d’angular qui récupere le Token stocké le local storage du navigateur l’envoi par requête afin de récupérer l’utilisateur qui correspond à ce Token

### 5.3.2 Le deployement

N’ayant jamais déployé une application en ligne, il m’a fallu faire pas mal de recherche afin de trouver les bonne informations pour ensuite les triés et d’en garder les plus pertinentes.  
La solution a été de suivre un tutoriel sur YouTube et de l’adapter a mon projet.

### 5.3.3 La lenteur

Déployé, l’application de part un abonnement Azure peu cher était extrêmement lente.   
Lors d’un chargement de page il fallait attendre la réponse de la requête pour afficher les données ce qui rendait la navigation dans l’application et l’expérience client mauvaise.  
La solution a été d’utiliser des observables dans la partie front-end afin d’y stocker les données.

Lors de la navigation ne plus faire des requête vers la base de données mais bien d’aller lire ces observables en en souscrivant (‘subscribe’)  
L’avantage de ceux-ci est que lors de chaque modifications les changements sont envoyer vers les vues qui sont dite abonnée (‘subscribe’) à cet observable.  
Dés lors les données peuvent être affichée en attendant la réponse de la requête. Des que celle si sera arriver la page sera automatiquement mise a jour

### 5.3.4 La latence

Déployée, le code créé pour la partie front-end m’a dévoilé ses faiblesses.   
En effet, lors de tests en local, aucune latence ne permettait de généré des bugs.

Par contre, une fois déployé, le temps entre l’envoie d’une requête et sa réponse permettait de cliquer plusieurs fois sur un même bouton. Dés lors, plusieurs requêtes identiques étaient envoyées au server ce qui le saturait, gênerait informations multiples en DB.  
La solution a été de créer une variable boolean pour chaque bouton dans la partie front-end appelée hiddenButton, de la passer a true lors du clic cette variable étant liés au bouton celui-ci n’est pas visible mais remplacé par un message adapté et lors de la réponse passer cette même variable a false pour revenir a l’état d’origine

### 5.3.5 L’utilisation d’une adresse SMTP

Pour la modification du mot de passe, j’ai créé une adresse mail à configurer en SMTP  
Pour cela j’ai utiliser Gmail grâce a ce tutoriel :   
[Expédier des emails via SMTP depuis votre compte Gmail - Centre d’aide PrestaShop](https://help-center.prestashop.com/fr/articles/11094179575698-expedier-des-emails-via-smtp-depuis-votre-compte-gmail)

Cela fonctionne parfaitement en local par contre lorsque l’application server est déployé sur Azure lien envoyé par e-mail n’arrive pas jusqu’à l’URL voulue  
Je n’ai malheureusement pas encore trouvé de solution a cela et suit toujours en recherche

### 5.3.6 Le temps

Le temps et l’organisation de celui-ci est un facteur prépondérant de la bonne réussite de ce projet.   
Il m’a fallu en consacré beaucoup et segmenter son l’utilisation afin de répondre a toutes les demandes

## 5.4. Formation et suivi

### 5.4.1 TechnofuturTic

Avant et pendant ma première année de bachelier j’ai eu la chance de faire 2 formations de 6 mois à Technofuturtic  
J’y appris les base du .NET et d’Angular. J’ai pu parfaire cette apprentissage en 2éme année dans le cours d’orienté objet ou nous avons codé en C# et lors du cours de projet cette dernière année ou j’ai pu réalisé un projet de gestion de restaurent avec les mêmes technologies que pour ce TFE  
L’apprentissage a donc été progressif en continu pour ces technologie

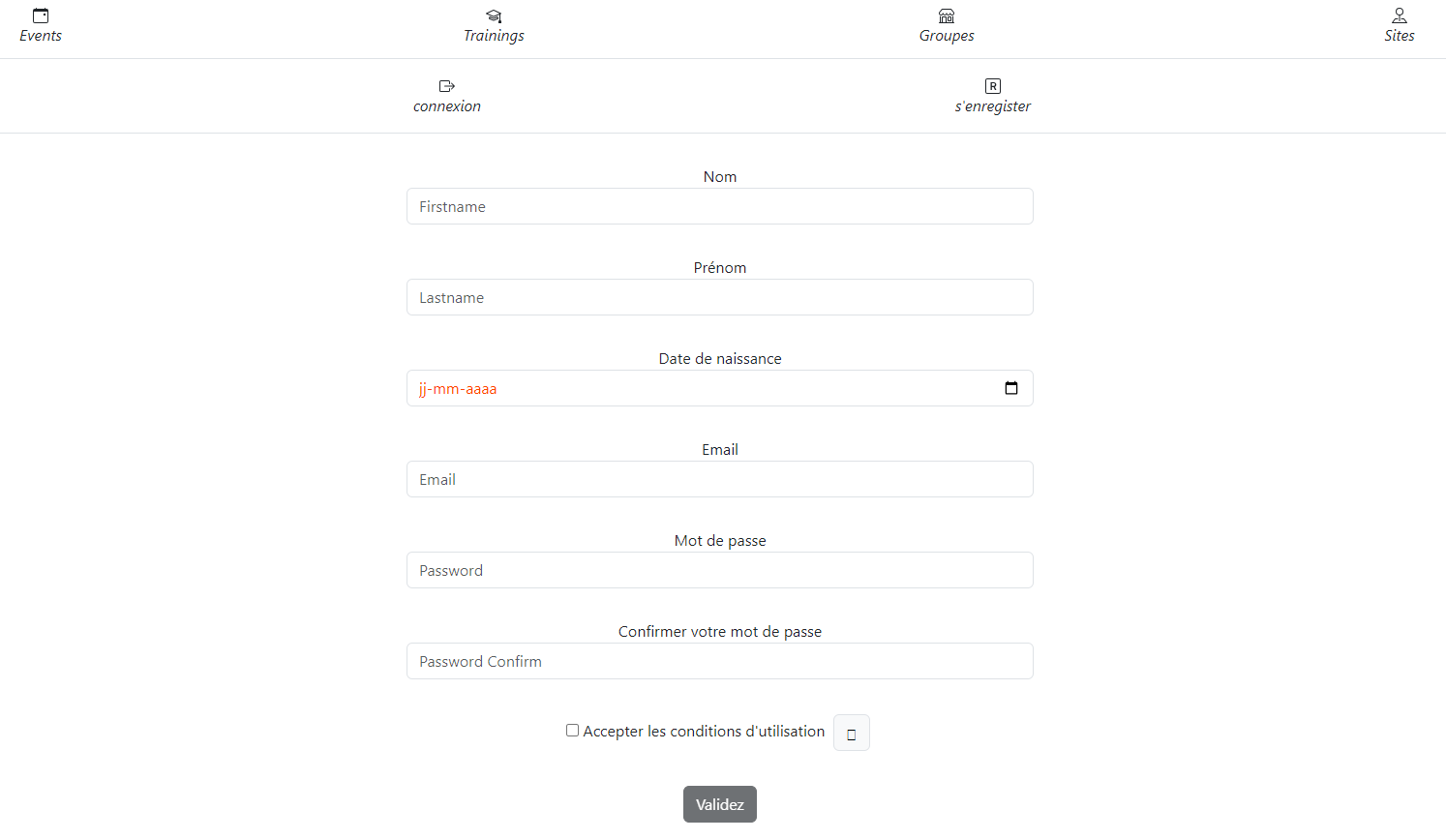
### 5.4.2 Transurb

1. Entre ma 2eme et 3eme année de Bachelier, j’ai la chance de travailler pour Transub  
   Transurb est une société belge proposant des solutions informatisées sur mesure en matière de transports ferroviaires et urbains.   
   J’ai pu au travers de différents projets développer mes aptitudes à coder en C# , c’est d’ailleurs là que j’ai fait mon stage.

## 5.5. Le résultat

### 5.5.1 Home

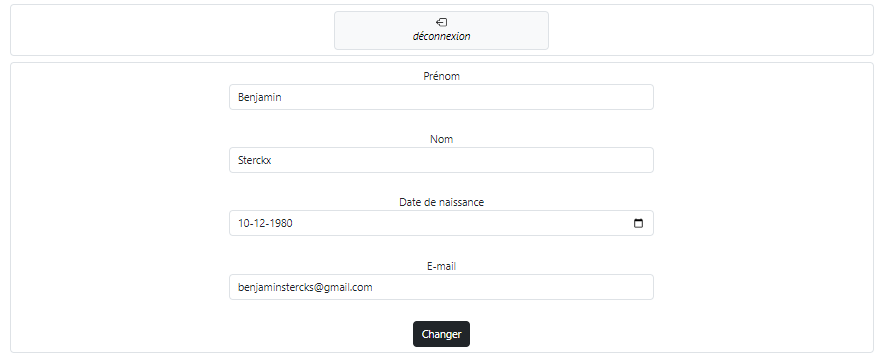
### 5.5.2 S'enregister

****

### 5.5.3 Connexion

### 5.5.4 Utilisateur connecté

### 5.5.5 Profil

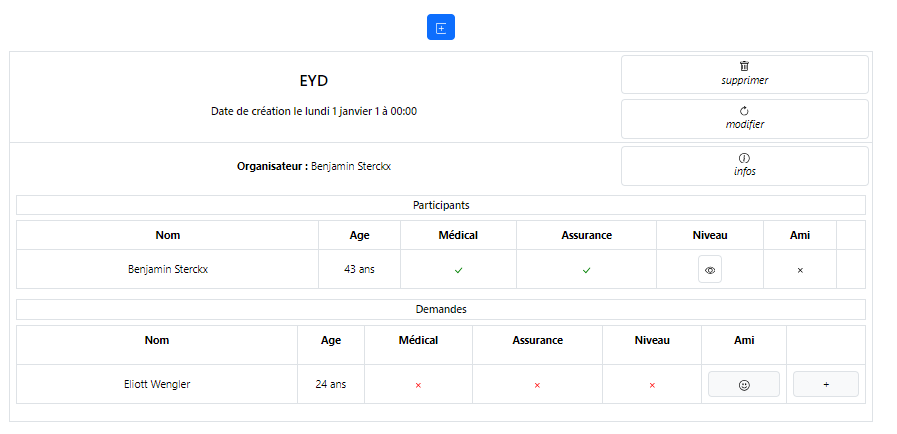
****

### 5.5.6 Contacts

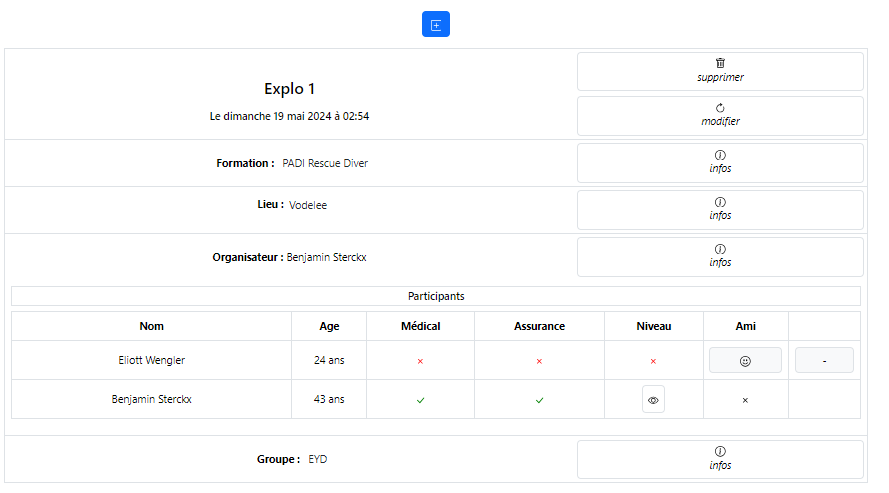
### 5.5.7 Les sites

****

### 5.5.8 Les groupes

****

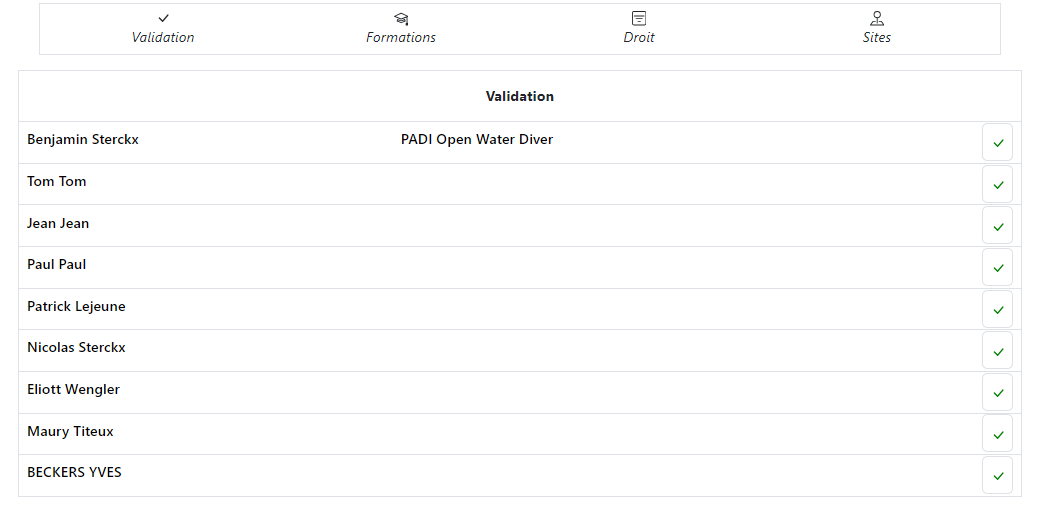
### 5.5.9 Les évènements

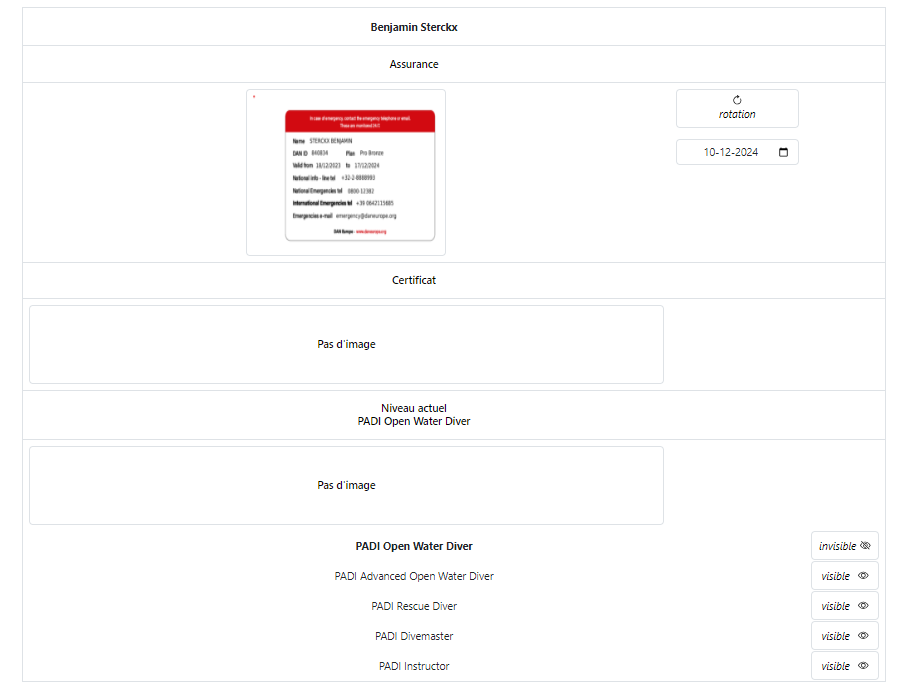
****

### 5.5.10 les messages

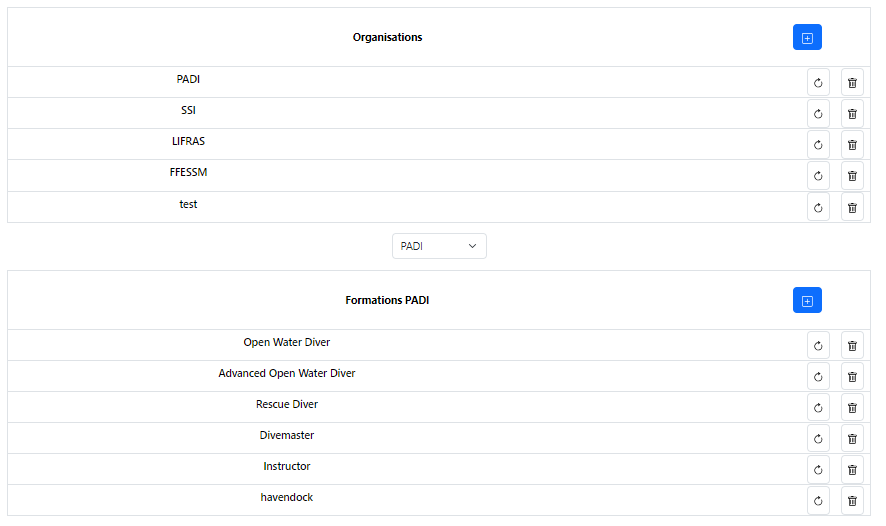
### 5.5.11 Admin

Validation

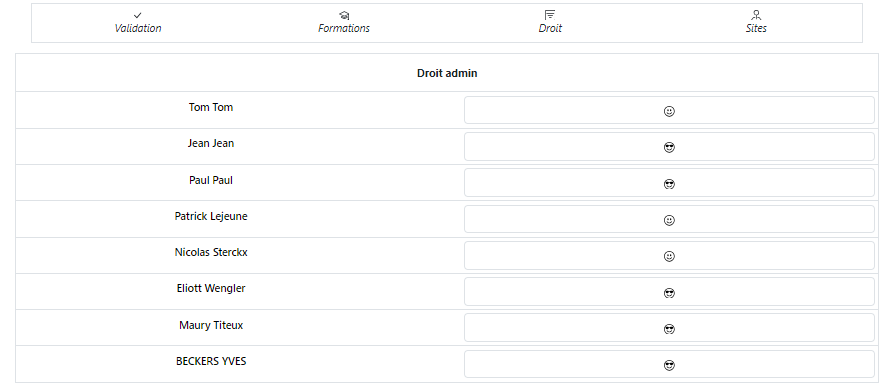
****

****

Formation

****

Droit

****

# 6. Evolution(s) immédiate(s) pressentie(s)

## 6.1 Augementer les adherants

Attirer de plus en plus d’utilisateurs en rendant l’application plus accessible et plus rapide.

Mettre en place une promotion ciblée sur Facebook dans les groupes de plongeurs en Belgique.

Faire valoir l’application lors de salons, de journées portes ouvertes, etc., dédiés à la plongée.

Mettre en place en référencement adapté.

## 6.2 Créer des nouvelles fonctionnalitées

Les retours des utilisateurs seront cruciaux pour cette partie afin de créer de nouvelles fonctionnalités qui répondent à une demande. Pour ce faire, il nous faudra recueillir les avis des utilisateurs de différentes manières, en fonction de leurs profils (utilisateur ou administrateur).

## 6.3 Monter une société

Nous estimerons le nombre d’adhérents, d’utilisateurs et de sponsors nécessaires afin d’établir un business plan. Celui-ci permettra de déterminer la faisabilité et le type de structure à créer, qu'il s'agisse d'une ASBL, d'une SA ou d'une SPRL.

## 6.4 Les ambassadeurs

Les ambassadeurs seront des utilisateurs qui faciliteront l’évolution et la promotion de l’application. Un système de notation sera mis en place pour chacun d’eux. Les meilleurs notes auront un meilleur référencement dans l’application. Par exemple, l’instructeur ayant la meilleure note aura ses formations affichées en premier.

## 6.5 Un design

Le design étant actuellement volontairement sobre et ne connaissant pas encore les attentes des utilisateurs celui-ci pourra être adapté progressivement afin que les utilisateurs s'y retrouve.  
Le boutons d’actions seront eux toujours tous de la même couleur pour que les actions reste intuitives pour l’utilisateur

## 6.6 Les invitations

Un système d’invitation était prévu mais n’a pu être mis en place par manque de temps.  
Celui-ci sera l’un des priorité futur et consistera a permettre d’inviter des amis dans les groupes ou événements mais encore de pouvoir inviter tout un groupe a un événements

## 6.7 Les messages par groupes

Ajouter la possibilité d’envoyer un message a tout un groupe est l’une des améliorations à l’étude, cette idée m’a été recommander les premiers utilisateurs de l’application

# 7. Évaluation du travail

## 7.1. Autocritique

### 7.1.1. L’élaboration du cahier des charges

Le fait d'être seul a développer m'a rendu l'élaboration du cahier des charges assez confortable seul les interviews sont venue aiguisé ce travail.

Cela a aussi peut être porté légèrement préjudice dans le sens où je modifiais à loisirs ce qui me semblait non approprié. Hors, le fait d'avoir des contraintes précises me force à être plus carré dans mes choix et à

moins me dissiper par la suite.

### 7.1.2. La conduite de l’analyse

Celle-ci s'est faite de manière itérative grâce à la méthode Agile.

Ayant une idée clair du résultat attendu, j’ai pu définir les priorités et segmenter le travail d’analyse par fonctionnalités.

Le fait de progresser de manière linéaire lors de chaque itération m’a permis utiliser les idées développées au point prétendant pour le point suivant et ce en commençant par un descriptif puis un scenario, diagramme d’activité et finir par le diagramme de classe.  
L’utilisation des outils informatique tels que Looping et Plant Uml on été un vrai atout pour réaliser des schémas clair.

### 7.1.3. La réalisation

La mise en place des méthodes Agile décrite ci dessus et Scrum m’ont permis de planifier grâce au logiciel MeisterTask les différentes taches a réaliser et tant bien que mal de respecter ce planning.  
L’utilisation d’outils déjà utiliser lors de projet précédant mon clairement avantagé dans l’évolution de ce travail

Néanmoins par manque de temps, j’aurais aimé présenter une solution plus aboutie et résoudre touts les problèmes rencontrés lors du déploiement. C’est en effet sur cette partie, n’ayant pas encore eu l’expérience de mettre une application en ligne, que j’ai du revoir ma copie.  
L’application était trop lente pour que l’expérience utilisateur soit confortable de plus le temps de requête n’avait pas été prise en compte lors de la conception en local.  
J’ai également afin d’avancer rapidement travaillé en MVP, Minimum Viable Product.  
Cela m’a permis d’avoir en rendu d’avoir rapide mais par le suite j’ai du revoir le travail réalisé, je suis mitigé sur cette méthode.

## 7.2. Appréciation(s) du client

Les retours sont positif quant aux idées mis en place, elles ont été en grande partie réalisées et le rendu leur semble attrayant.

Avoir une tel application qui facilite l’organisation de plongées peut, selon eux, trouver sa place permis celle existante car aucunes n’est orienté 100% plongée.

Néanmoins là ou le bas blesse c’est la vitesse d’exécution, l’expérience utilisateur et j’y travail d’arrache pied

# 8. Conclusion

Mettre en œuvre l’apprentissage de 3 ans études a été salutaire. Ce travail est déjà un bel accomplissement.

Celui-ci repose sur une analyse orientée base de données et Merise mais également sur une analyse UML des besoins de l’application.

Les potentielles évolutions futures de l’application m’ont permis de revoir en détails mes cours de communication, de e-business et de droit.

Les cours de développement et d’analyse m’ont quant a eux permis de choisir des outils de programmation et une méthodologie qui m’ont permis d’aboutir à un résultat.

L’objectif était de mettre en ligne, une premier version de l’application qui répond aux besoins des interviewés.  
Cet objectif été atteint puis que l’heure ou j’écris ces lignes, il y a déjà 5 personnes qui s’y sont inscrite et peut être plus d’ici la présentation soyons optimiste.

A titre personnel, je suis satisfait du résultat et pense qu’à terme, cette application peut réellement trouver sa place dans le monde de la plongée.

Ce projet selon moi n’est qu’a ses débuts, mon travail de fin d’étude n’a été que le moteur pour mettre en place une idée qui me tient a cœur.

L’utilisation de framework fort demandé sur le marché de l’emploi tel Angular et .NET ne peut être que bénéfique pour ma futur recherche d’emploi.

Il va de soi que cette application est encore perfectible, elle a été réalisée par une personne

seule et à temps partiel, qui devait combiner ce travail avec énormément d’autres choses. Néanmoins,

les perspectives d’évolutions futures présentées plus haut prouvent qu’un bel avenir pourrait s’offrir

à elle !

# 9. Bibliographie

# 10. Annexes

## 10.1. Les scripts SQL

## 