

# Rapport de stage

Alexis Lapeze

BTS SIO 2<sup>ème</sup> année

Institut Limayrac



# Sommaire :

## Table des matières

1. Remerciements :	3
2. L'entreprise :	4
a. Présentation de l'entreprise :	4
b. Localisation de l'entreprise :	5
3. Contexte technique :	6
a. Introduction :	6
b. Technologies étudiées :	6
c. Le MVVM (Modèle Vue Vue Modèle) avec Angular :	7
d. La POC (Programmation Orienté Composant) :	8
e. Le Versioning avec Git et Gitlab :	9
4. Missions réalisées :	10
a. L'application entretien annuel :	10
b. Cahier des charges :	10
c. Base de données :	11
d. Maquettage et IHM :	12
5. Conclusion :	13
6. Annexes :	14
a. Liste des taches / fonctionnalités à réaliser :	14
b. Extrait de la JDL :	15
c. Maquettes :	16

# I. Remerciements :

Je tiens à remercier tout particulièrement :

- Mathieu Monin, responsable du pôle digital de l'entreprise et tuteur de mon stage, pour son aide, ses conseils et son temps lors de mes différentes phases de questionnements, pour toutes les réponses qu'il a pu m'apporter et l'aide qu'il a pu me fournir.
- Joffrey Hernandez, Alexandre Merignac et Pierre Lafont développeurs et consultant au sein d'Inovans, pour tous leurs conseils et leurs astuces de développement apporter lors de mes phases de développements.
- Jenna Dugain, graphiste de l'entreprise, pour ses maquettes et ses conseils en matière d'expérience utilisateur.
- Tous les employés d'Inovans, pour leur accueil et leur bonne humeur.

## 2. L'entreprise :

### a. Présentation de l'entreprise :

Inovans est une société de services (ESN) créée en 2001 et rachetée en 2011 par le fond d'investissement Alan Allman Associates. Elle est spécialisée dans le secteur de l'activité du conseil en systèmes et logiciels informatiques. Inovans travaille avec des entreprises de différents types : Start-ups et grands comptes.

Comme toutes les sociétés de services, Inovans propose son expertise informatique aux entreprises dans leur transformation digitale ou l'amélioration et la maintenance de logiciels existants.

Appartenant à Allan Allman Associates, Inovans profite de son écosystème qui lui permet d'avoir un financement régulier afin de pouvoir répondre rapidement et facilement aux besoins spécifiques de ses clients.



## **b. Localisation de l'entreprise :**

L'entreprise est située à Toulouse au 2 rue du professeur Pierre Vellas. Elle est idéalement située, car proche des grandes entreprises du secteur aéronautique et de leurs sous-traitant avec qui elle travaille.



## 3. Contexte technique :

### a. Introduction :

Mon rôle au sein de l'entreprise était de poursuivre le développement d'une application interne dont le développement avait à peine commencé. Cette application leur permettrait de gérer facilement et rapidement les bilans annuels de leurs consultants souvent en mission à l'extérieur des locaux de l'entreprise. Cette application sera incluse dans un panel d'application, aujourd'hui toujours en cours de développement, facilitant l'interaction entre les consultant et les managers de l'entreprise.

### b. Technologies étudiées :

Ma période de formation à commencée avec une phase de montée en compétences d'environ deux semaines sur les technologies utilisées pour le développement des applications de l'entreprise. Les applications d'Inovans sont découpées en deux parties : un backend (arrière-plan) et un frontend (avant-plan).

Le backend est le même pour tout le panel d'application de l'entreprise. C'est une API (« Applicative Programming Interface » Interface de Programmation Applicative) connectée à la base de données de l'entreprise (PostgreSQL). C'est un projet généré avec le générateur JHipster permettant d'élaborer des applications rapidement.

JHipster (pour Java Hipster) fournit des outils pour générer un projet Spring Boot (le backend) et un frontend avec Angular et Bootstrap. (Pour les besoins de l'entreprise, un travail de dissociation a été effectuer afin de ne garder que la partie backend).

JHipster génère automatiquement la base de données par le biais de d'un langage spécifique (JDL), créé des méthodes CRUD pour chacune des tables de la base de données et facilite le traitement de ces données par le biais de DAO / DTO.

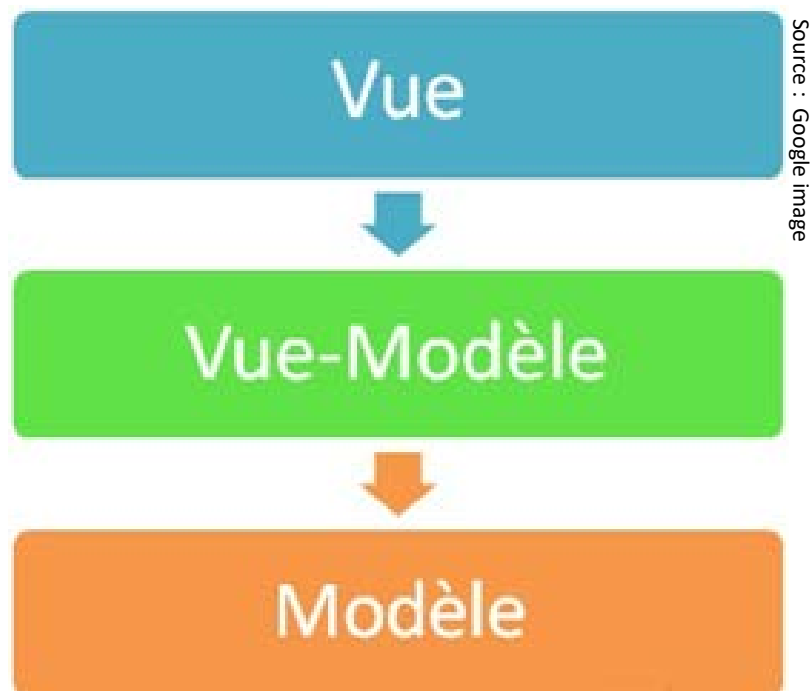
Tous les fichiers sont générés sur le modèle d'une application Spring Boot. C'est un framework JAVA dont le but est de faciliter et de rendre productif le développement d'applications.

Le frontend de chaque application est basé sur le framework Angular. Développé par Google, propose un typage accru des données grâce au Typescript qui vient dans le développement remplacer le JavaScript ainsi qu'un développement orienté composant. Les pages web ne sont donc plus vue comme des blocs uniques, mais découpées en « composant » qui peuvent être réutilisé dans de multiples pages.

Angular propose par rapport aux autres framework JavaScript une architecture plus encadrée, ainsi il n'y a qu'une seule façon de faire telle ou telle fonctionnalité ce qui permet aux développeurs de retrouver la même architecture dans n'importe quel projet Angular. Le choix de cette technologie a été approuvé du fait que toutes les applications de l'entreprise allaient être basées sur le même « template » qui est lui-même un projet de l'entreprise. Ce projet consiste à créer un socle de développement commun à toutes les applications afin d'en réduire le temps de développement et la maintenance.

### c. Le MVVM (Modèle Vue Vue Modèle) avec Angular :

Le MVVM est un pattern (patron de conception) qui a été spécialement conçu pour améliorer la séparation entre les données et la vue qui les affiche. Le lien entre la vue et le modèle de données est fait par des mécanismes de binding (un binding est un mécanisme qui permet de faire la liaison entre des données de manière dynamique). En finalité, ce qui est modifié dans le modèle se répercute dans la vue de façon automatique et dynamique.



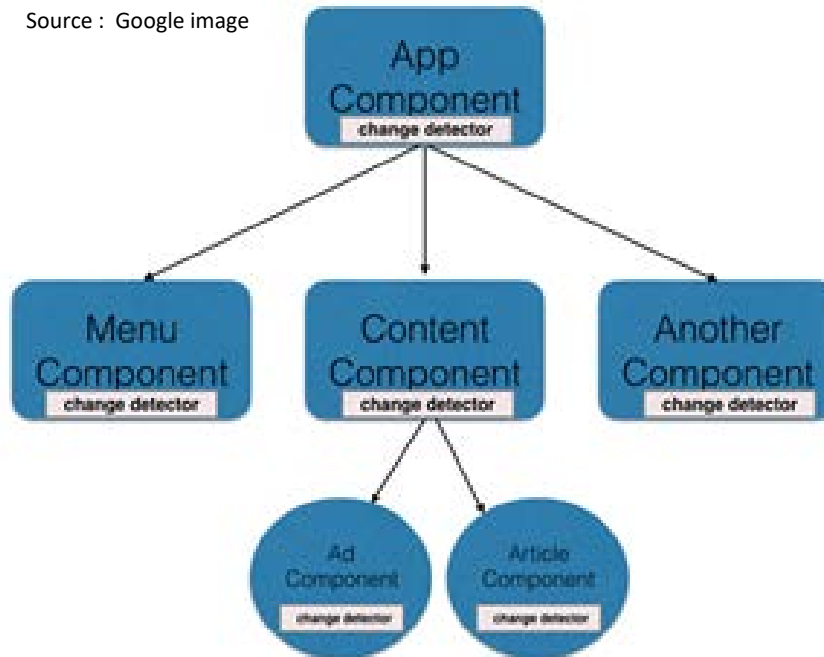
Dans le principe du MVVM, le Modèle contient les données, la Vue est ce qui est affiché à l'utilisateur, et la Vue-Modèle s'occupe de faire le lien entre le Modèle et la Vue. C'est ici qu'intervient le binding. Dans ce cas, la Vue ne doit jamais traiter les données, mais uniquement se charger de les afficher à l'utilisateur. Le Vue-Modèle aura en charge les conversions et les accès au modèle.

En conclusion, le MVVM, se repose sur le DataBinding et l'événementiel. La vue est couplée aux données (stocker dans le Modèle) via les mécanismes de binding et invoque les méthodes du Vue-Modèle. Le Vue-Modèle lui invoque les méthodes du modèle qui contient les données.

Angular étant malgré tout une technologie JavaScript, il n'utilise pas le DOM (Document Object Model) comme les technologies javascript standards, mais le MVVM qui est une méthode beaucoup plus structurée.

#### d. La POC (Programmation Orienté Composant) :

Source : Google image



Le principe de la programmation orientée composant est de rendre réutilisable une fonctionnalité à l'ensemble du projet. Cela permet d'alléger le projet lorsqu'une fonctionnalité est appelée à se retrouver à plusieurs endroits dans le projet et de rendre la maintenance du site web plus simple.

Un composant peut être comparé à une fonction appelée plusieurs fois dans le même code. Avec Angular, c'est le même principe. Comme le montre le schéma ci-dessus, il y a un composant père (App Component) qui contient tous les composants de l'application et des composants enfants qui correspondent à une partie du site (ex : le composant « Menu »). Ainsi, le projet est découpé en différents composants distincts. Chaque composant nécessite une ou plusieurs fonctionnalités. Par exemple, le composant « Content » contient les composants « ad » et « article ».



## e. Le Versioning avec Git et Gitlab :



Git est un logiciel de gestion de version décentralisée. Créé en 2005 par Linux Torvald (créateur du noyau Linux), il est open-source et permet de pouvoir collaborer à plusieurs sur un seul et même projet en évitant les conflits de modification d'un même fichier dans un projet par plusieurs développeurs.



Gitlab est un service web d'hébergement et de gestion de développement logiciel utilisant le logiciel de gestion de version Git.

L'entreprise utilise Gitlab pour pouvoir y stocker et y sauvegarder différentes versions de leurs logiciels en développement. Pour chaque projet, Inovans travail avec plusieurs branches :

- Une branche « Production » qui est la branche avec l'état actuel de l'application en ligne utiliser par les utilisateurs. Elle peut être comparée à la branche « Master » qui est la branche principale de l'application.
- Une branche « Develop » qui centralise toutes les fonctionnalités de la version suivante de l'application.
- Une branche « feat/nom\_de\_la\_feature » qui correspond à une fonctionnalité bien précises de l'application.

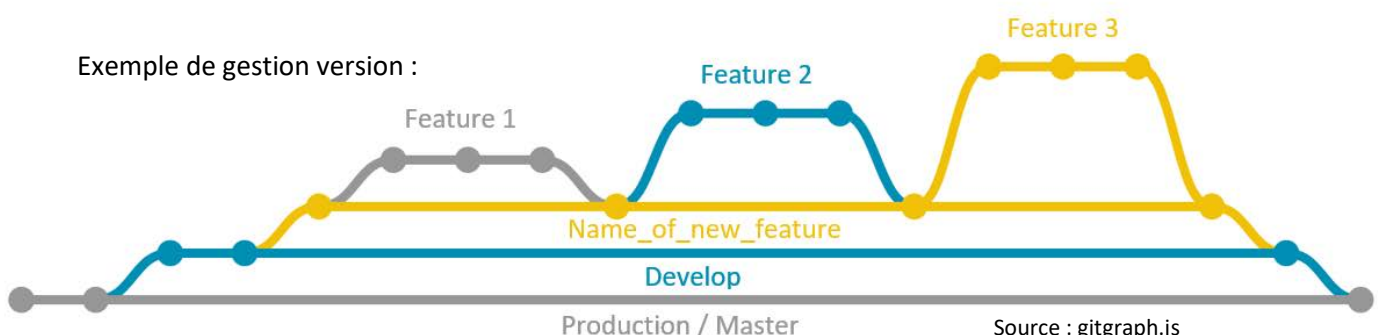
Gitlab permet facilement de créer des « Merge request » ainsi que des « Issues ».

Les « Merge request » (pour « Demande de Fusion ») permettent de soumettre la fusion d'une branche à une autre. Ainsi, lorsqu'une fonctionnalité est terminée, le développeur de cette « feature » crée une « Merge request » de sa branche à la branche « Develop » et un autre développeur vérifie et valide l'intégration du nouveau code à celui de la branche « Develop ».

Les « Issues » permettent aux développeurs de lister et de suivre l'évolution d'un bug trouvé ou d'une proposition d'amélioration possible.

Avec ce mode de fonctionnement, l'entreprise assure une certaine qualité dans le développement de l'application et diminue son temps de développement ce qui a un impact favorable sur le coût de l'application.

Exemple de gestion version :



## 4. Missions réalisées :

### a. L'application entretien annuel :

Lors de cette période de formation, j'ai été amené à poursuivre le développement d'une application web pour les besoins de l'entreprise. Comme dit précédemment, cette application a pour but de permettre aux managers de l'entreprise de faciliter la préparation de l'entretien annuel de chaque consultant. Ces derniers étant souvent en mission à l'extérieur des locaux de l'entreprise, il leur est très difficile de préparer cet entretien par téléphone ou par mail.

Cette application a aussi pour but de permettre à n'importe quel consultant de pouvoir visualiser la date d'anniversaire des autres consultants et de pouvoir proposer des idées de cadeaux à ces derniers.

L'application comporte aussi une hiérarchie, ainsi les utilisateurs ayant un rôle administrateur ont accès à une liste de fonctionnalités comme par exemple avoir accès à la liste des bilans annuels effectués d'un utilisateur.

L'entreprise m'a donc fourni un fichier PDF correspondant au formulaire à remplir au préalable avant la date de l'entretien. Il m'a donc été demandé d'intégrer ce formulaire à l'application et d'en gérer les accès et les droits.

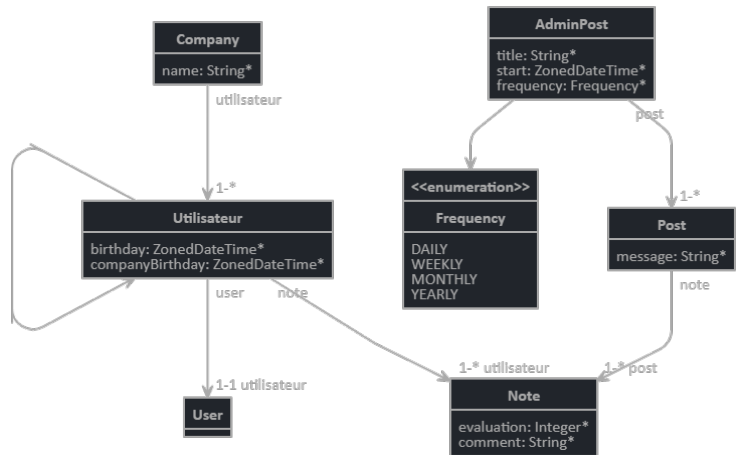
### b. Cahier des charges :

Fonctionnant avec des méthodes plutôt agiles, le cahier des charges n'était pas explicitement fourni sous un document papier. Cependant, les grandes fonctionnalités à réaliser ont été placées dans les « Issues » du projet sur le Gitlab.

### c. Base de données :

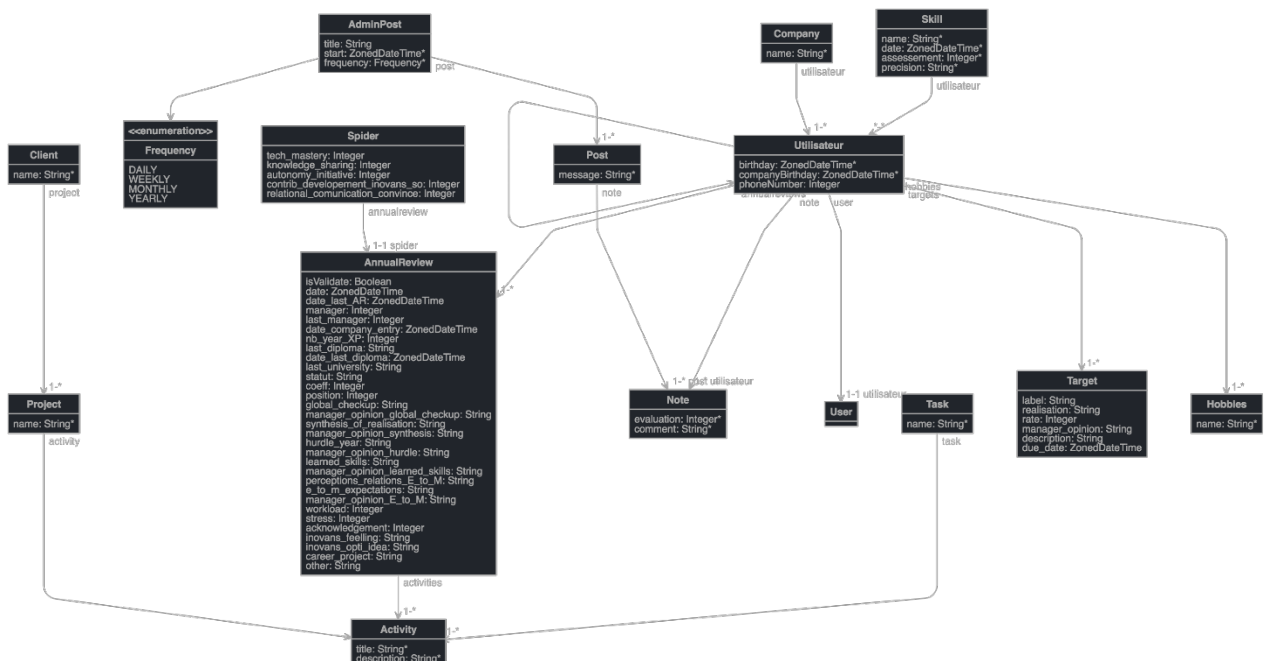
N'ayant pas créé cette application de zéro, afin d'y intégrer les évolutions que j'ai apportées au projet, j'ai dû aussi faire évoluer le schéma de la base de données.

JHipster englobant la base de données, toutes modification de celle-ci passe par le biais d'un langage spécifique : [JDL \(JHipster Domain Language\)](#). JHipster fourni un outil en ligne permettant de créer sa base de données et d'en avoir la modélisation au format diagramme de classe. (JDL Studio)



Base de données existante

Comme le montre le diagramme ci-dessous, un utilisateur possède un ou plusieurs « AnnualReview » (bilan annuel). Ce bilan annuel est lui-même composé de différentes tables.



Evolution de la base de données

Après avoir modélisé la base de données avec l'outil fourni, l'importation au projet se fait via une commande « `jhipster import-jdl ./jhipster-jdl.jh` » (il faut au préalable avoir copier-coller dans le fichier « `jhipster-jdl.jh` » le code de la JDL). Cette commande crée donc les différentes tables dans la base de données, mais crée aussi dans le projet Spring les différentes classes, DTO, ainsi que les services et les ressources REST avec les méthodes CRUD de chacune des classes.

#### **d. Maquettage et IHM :**

Par souci de confort et afin de réduire le temps de développement, le pôle design de l'entreprise à réaliser des [maquettes](#) de l'interface utilisateurs de l'application. Étant intégré pleinement dans le projet, mon avis a été sollicité et j'ai donc apporté mon point de vue et fait des propositions d'améliorations aux maquettes existantes ainsi qu'aux fonctionnalités prévues et possibles à l'application.

## 5. Conclusion :

Lors de cette période de formation, j'ai eu l'occasion suivre une veille technologique constante en lisant les documentations sur les différentes technologies que j'utilisais. Ce stage m'a donc permis d'enrichir mes connaissances techniques mais aussi théoriques sur différentes techniques de programmations. Travailler sur un projet réel et concret apporte énormément de rigueur dans les tâches à réaliser, mais aussi beaucoup de compréhension dans la nécessité de l'apprentissage par le biais d'études ou de l'autodidactie pour porter a bien un projet quel qu'il soit.

J'ai pu poursuivre mon apprentissage dans le domaine de la technique, mais aussi apprendre dans le domaine des sociétés de services et de la gestion de projet lors des différentes discussions que j'ai pu avoir avec les collaborateurs de l'entreprise. Cela m'a permis de comprendre plus clairement les points abordés en cours sur l'économie, le droit et le management d'entreprise.

## 6. Annexes :

### a. Liste des taches / fonctionnalités à réaliser :

<b>Sauvegarde des bilans précédents</b> #11 · opened 4 weeks ago by Dugain ⓘ Pouvoir remplir mon bilan annuel
<b>Pouvoir remplir mon bilan annuel</b> #10 · opened 4 weeks ago by Dugain ⓘ Pouvoir remplir mon bilan annuel
<b>Programmer un rendez-vous pour le bilan avec le consultant</b> #9 · opened 4 weeks ago by Dugain ⓘ En tant que BM/Admin avoir accès au bilan d'une personne
<b>Voter pour une idée de cadeau</b> #8 · opened 4 weeks ago by Dugain ⓘ Avoir accès aux anniversaires de mes collègues
<b>Proposer une idée de cadeaux</b> #7 · opened 4 weeks ago by Dugain ⓘ Avoir accès aux anniversaires de mes collègues
<b>Export .docx</b> #6 · opened 4 weeks ago by Dugain ⓘ En tant que BM/Admin avoir accès au bilan d'une personne
<b>Export pdf bilan</b> #5 · opened 4 weeks ago by Dugain ⓘ En tant que BM/Admin avoir accès au bilan d'une personne
<b>Profil d'un(e) collègue</b> #4 · opened 4 weeks ago by Dugain ⓘ Avoir accès aux anniversaires de mes collègues
<b>Rechercher une personne</b> #3 · opened 4 weeks ago by Dugain ⓘ Avoir accès aux anniversaires de mes collègues
<b>Liste anniversaire</b> #2 · opened 4 weeks ago by Dugain ⓘ Avoir accès aux anniversaires de mes collègues
<b>Filtres</b> #1 · opened 4 weeks ago by Dugain ⓘ Avoir accès aux anniversaires de mes collègues

Source : Gitlab Inovans

## b. Extrait de la JDL :

```
1 // Globals
2 DEFAULT_MIN_LENGTH = 0
3 DEFAULT_MAX_LENGTH = 255
4
5 // Enums
6 enum Frequency {
7     DAILY, WEEKLY, MONTHLY, YEARLY
8 }
9
10 // Domain
11 entity Activity {
12     title String required,
13     description String required minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH)
14 }
15
16 entity AdminPost {
17     title String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
18     start ZonedDateTime required,
19     frequency Frequency required
20 }
21
22 entity AnnualReview {
23     isValidate Boolean,
24     date ZonedDateTime,
25     date_last_AR ZonedDateTime,
26     manager Integer,
27     last_manager Integer,
28     date_company_entry ZonedDateTime,
29     nb_year_XP Integer,
30     last_diploma String,
31     date_last_diploma ZonedDateTime,
32     last_university String,
33     statut String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
34     coeff Integer,
35     position Integer,
36     global_checkup String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
37     manager_opinion_global_checkup String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
38     synthesis_of_realisation String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
39     manager_opinion_synthesis String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
40     hurdle_year String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
41     manager_opinion_hurdle String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
42     learned_skills String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
43     manager_opinion_learned_skills String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
44     perceptions_relations_E_to_M String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
45     e_to_m_expectations String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
46     manager_opinion_E_to_M String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
47     workload Integer min(0) max(10),
48     stress Integer min(0) max(10),
49     acknowledgement Integer min(0) max(10),
50     inovans_feelling String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
51     inovans_opti_idea String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
52     career_project String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH),
53     other String minlength(DEFAULT_MIN_LENGTH) maxlength(DEFAULT_MAX_LENGTH)
54 }
55
```

## c. Maquettes :

