

# Touffu

Projet annuel (2022) - ESGI 3e année - Architecture Logicielle

Nathan LETOURNEAU

Étudiant en Architecture Logicielle

École Supérieure de Génie Informatique

@Nathan-dev-dot

nathan.letourneau@net-c.com

Théo OMNES

Étudiant en Architecture Logicielle

École Supérieure de Génie Informatique

@NightTheo

omnes.theo@gmail.com

Sarah SCHLEGEL

Étudiante en Architecture Logicielle

École Supérieure de Génie Informatique

@SarahSch19

sschlegel@protonmail.ch

**Abstract**—Le projet annuel de la filière Architecture Logicielle de l'ESGI a pour but de rassembler en un seul projet la plupart des compétences acquises au cours de l'année. En 2022, il s'articule autour des éléments suivants : réalisation d'un front-end en AngularJS, d'une application lourde en Java, d'une API en NodeJS, d'une base de données noSQL, d'une base de données relationnelle, et de la création d'un langage de scraping. Axé sur le thème de l'aide à la personne, il propose de combiner les éléments précédents afin de concevoir un service numérique complet.

## I. INTRODUCTION

Touffu est une application d'aide à la personne, centrée sur les soins et promenades d'animaux de compagnie pour les personnes dépendantes [1]. Le service a pour but de mettre en relation les personnes dépendantes, nécessitant de l'aide, avec des volontaires déclarés qui peuvent s'occuper des animaux domestiques [2].

Au niveau technique, le service se divise en sept modules majeurs :

- 1) Une application client en AngularJS, destinée aux utilisateurs principaux du service : elle permettra aux utilisateurs de s'inscrire et de trouver ou de proposer leurs services en fonction de leur catégorie ;
- 2) Une application d'administration en AngularJS, destinée aux gérants de l'application, afin de gérer les comptes des utilisateurs, la base de données, les catégories principales de l'application, etc. ;
- 3) Une API en NodeJS pour faire la jonction entre les deux applications front-end et la base de données ;
- 4) Une base de données en noSQL, accessible à partir de l'API ;
- 5) Une application lourde de gestion du projet Touffu en Java destinée aux développeurs de Touffu. Cette application doit notamment inclure une messagerie interne et une gestion de plugins.
- 6) Une base de données relationnelle pour gérer les données associées au pilotage du projet, qui ne sera associée qu'à l'application Java.
- 7) Un langage de scraping en ligne de commande permettant de récupérer des informations sur les prestations procurées (soins et promenades des animaux de

compagnie pour les personnes dépendantes) à partir de différentes sources Web, pour les traiter et les enregistrer dans des fichiers.

Ces modules sont représentés de manière graphique dans la figure 1.

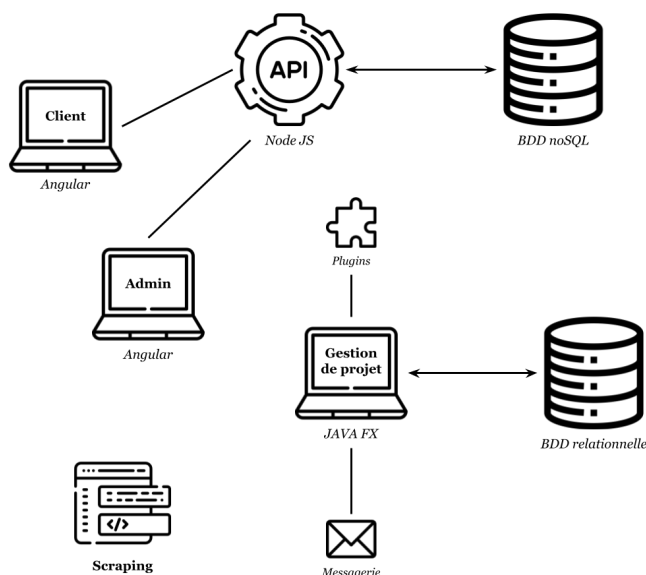


Fig. 1. Architecture globale

### A. Descriptif du projet



Fig. 2. Logo Global

### 1) Touffu client ("Touffu"):

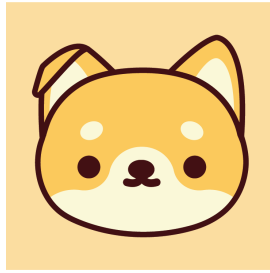


Fig. 3. Touffu

Du côté front-end, en s'inscrivant sur Touffu, les utilisateurs définissent s'ils souhaitent être aidés (bénéficiaires ou personnes tierces demandant en leur nom) ou apporter de l'aide (prestataires). L'application leur permet ensuite de définir leurs besoins (promenade, brossage, préparation de nourriture, changement de litière, accompagnement chez le vétérinaire) ou leurs disponibilités (types d'animaux, activités, possibilité d'effectuer des prestations récurrentes ou ponctuelles).

Les bénéficiaires qui ont besoin d'aide peuvent définir si elles ont besoin d'une aide récurrente, auquel cas elles peuvent définir une récurrence quotidienne, hebdomadaire ou mensuelle par exemple, ou d'une intervention ponctuelle. Ils peuvent définir si cette demande concerne plusieurs animaux ou un seul, et à combien de temps ils estiment cette prestation. Dans le cadre d'une intervention chez le vétérinaire, ils définissent également s'ils peuvent fournir un moyen de transport ou payer les frais de transport et comment ils couvrent les frais du vétérinaire.

Le système montre ensuite à chacun des partis une liste de personnes qui correspondent à leurs besoins et leurs disponibilités. Un système de messagerie interne leur permet d'échanger au préalable s'ils souhaitent dialoguer. Un calendrier propre à chaque utilisateur rappelle ensuite l'évènement à l'utilisateur.

Le prestataire (ou aidant) peut de son côté travailler pour plusieurs bénéficiaires à la condition que ses activités chez différentes personnes ne se chevauchent pas. Il peut également exercer plusieurs activités (qui peuvent concerner des animaux différents) chez un seul bénéficiaire sur une même plage horaire.

Un système d'évaluation permet de noter la prestation d'un prestataire par le bénéficiaire, et inversement le comportement du bénéficiaire envers le prestataire. Cette évaluation est prise en compte dans la mise en avant des prestataires et des bénéficiaires dans les recherches. Elle est également mise en évidence dans les statistiques individuelles de chaque utilisateur, qui affichent donc en plus de sa note ses performances individuelles.

La plateforme Touffu permet de gérer en ligne les paiements des prestataires (via Stripe ou système similaire), ce qui permet également de générer les factures des prestataires. Les prestataires et les bénéficiaires peuvent se mettre d'accord sur la fréquence des facturations et des paiements (paiements à la prestation, hebdomadaires ou mensuels). En fonction du mode de paiement défini, le système affichera un rappel pour le bénéficiaire de la prestation et enverra une notification (mail) qui l'amènera au système de paiement.

Si le bénéficiaire est en retard sur ses paiements, des sanctions progressives seront appliquées (suspension des prestations en cours, blocage de nouvelles annonces, verrouillage du compte).

### 2) Front admin ("TouffAdmin"):

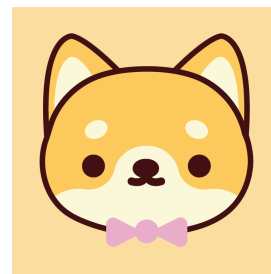


Fig. 4. TouffAdmin

Le front d'administration servira au service de Touffu pour la gestion de l'application. Cela comprend notamment la gestion des comptes utilisateur, la gestion des offres et demandes (suppression d'arnaques, gestion et envois des factures), de vérifier les messages qui ont été signalés par les utilisateurs comme inappropriés.

Ce backend permet également de gérer les comptes administrateurs (ajouter ou supprimer des administrateurs ou des modérateurs, définir leurs droits individuels ou de groupe).

Le front client est également géré à partir du backend. Cela inclut de gérer l'apparence du front client (images sur les sliders, texte de présentation, titres, etc.) et les données dans les formulaires (contenu des listes des services disponibles).

Des statistiques sont générées régulièrement (tâche automatisée) pour évaluer les tendances dans les activités des prestataires et les demandes des bénéficiaires, et pouvoir afficher les tarifs moyens pour une prestation donnée.

Touffu prélève une commission sur le paiement en ligne, tant du côté du prestataire que du côté du bénéficiaire, afin de subvenir à ses propres besoins. Le pourcentage de cette marge est défini dans l'administration de Touffu.

### 3) Gestion de projet ("Touffu-Management"):

L'application de management est destinée aux développeurs de Touffu, afin qu'ils puissent préparer les différentes étapes

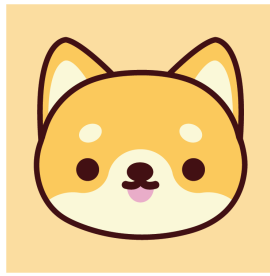


Fig. 5. Touffu Management

du projet, ainsi que les ressources humaines associées. Le management se fait via une interface de type Trello, avec une gestion de tableaux (cf. figure 6), colonnes et de cartes, également disponible en ligne de commande. Les utilisateurs peuvent ajouter, supprimer ou modifier des tableaux, cartes ou colonnes, et déplacer les cartes entre les colonnes.

Pour l'affectation des tâches, les utilisateurs peuvent s'affecter une tâche, ou l'affecter à un autre utilisateur présent sur le tableau.

Un transfert de droits sur un tableau peut avoir lieu si le propriétaire du tableau souhaite passer la main à un autre utilisateur. C'est le propriétaire d'un tableau (à l'origine son créateur donc) qui peut ajouter des membres à un tableau et gérer le tableau (suppression notamment).

Un gestionnaire de plugins intégré permet d'ajouter des plugins afin de rajouter des fonctionnalités aux tableaux. Par exemple, un plugin pourrait permettre d'ajouter des images à une carte, ou bien d'envoyer des notifications par email lors de la mise à jour d'une carte, ou encore d'ajouter des commentaires sur une carte.

Une messagerie interne permet aux utilisateurs de communiquer sur l'avancée du projet. Cette messagerie est propre à chaque utilisateur, et lui permet d'échanger avec un autre des utilisateurs de l'application.

Enfin, un système d'export permet d'exporter un tableau au format PDF.

#### 4) *Scraper ("Touffu Scrap"):*

Le langage compilé de Touffu permettra dans la mesure de raisonnable de scraper toutes les données du site client Touffu et d'autres sites similaires (récupérer des types de balises, les valeurs qu'elles contiennent...).

#### B. *Fonctionnalités optionnelles*

- Application mobile (Swift, "TouffApp") avec géolocalisation pour les promenades des animaux
- Système d'abonnement payant à Touffu pour les prestataires leur donnant des accès à des fonctionnalités plus pointues (liste non-exhaustive : statistiques détaillées, export des calendriers)

## II. FONCTIONNALITÉS

### A. *Interface utilisateur / Touffu*

#### 1) *Gestion des utilisateurs:*

##### **Inscription**

- Affichage de la page d'inscription
- L'utilisateur renseigne son email (doit être unique dans l'instance de la base de données), un mot de passe et son nom complet (nom, prénom), et sélectionne s'il souhaite demander de l'aide ou en proposer
- L'application vérifie l'unicité de l'email et la complexité du mot de passe (longueur supérieure à 8 caractères, au moins une majuscule, une minuscule, un chiffre et un caractère spécial)

Cas 1 : L'email est déjà utilisé

- Affichage d'un message d'erreur

Cas 2 : L'email n'est pas utilisé

- Affichage des champs d'inscription détaillés en fonction de la catégorie sélectionnée par l'utilisateur

Cas 2.1 : L'utilisateur propose son aide

- L'utilisateur sélectionne les animaux dont il souhaite s'occuper, ses types de prestations (récurrentes ou ponctuelles), ses disponibilités (jours et heures dans la semaine) et son rayon d'action géographique

Cas 2.2 : L'utilisateur a besoin d'aide

- L'utilisateur définit son emplacement géographique et liste son ou ses animaux
- Pour chaque animal, il définit les prestations nécessaires (promenade, brossage, préparation de nourriture, changement de litière, accompagnement chez le vétérinaire) et leur fréquence (quotidienne, hebdomadaire, mensuelle, ponctuelle)
- Dans le cas d'une prestation récurrente, l'utilisateur définit s'il le souhaite une date de début et de fin de la récurrence

Cas 2.1 et 2.2 - terminaison :

- Le système enregistre les données et crée le compte de l'utilisateur
- Le système transfère l'utilisateur à la page de connexion en affichant un message de succès

##### **Connexion**

- Affichage d'une fenêtre de connexion
- L'utilisateur entre ses identifiants de connexion (email et mot de passe)
- L'application vérifie la combinaison des identifiants

Cas 1 : Combinaison invalide

- La fenêtre affiche un message d'erreur

Cas 2 : Combinaison valide

- L'utilisateur est redirigé vers la page d'accueil de l'espace

utilisateurs

### **Gestion du profil**

- Affichage de la page de profil de l'utilisateur

Cas 1 : L'utilisateur est un prestataire

- L'utilisateur modifie les animaux dont il souhaite s'occuper, ses types de prestations (récurrentes ou ponctuelles), ou ses disponibilités (jours et heures dans la semaine)

Cas 2 : L'utilisateur est un bénéficiaire

- Pour chaque animal, l'utilisateur définit les prestations nécessaires (promenade, brossage, préparation de nourriture, changement de litière, accompagnement chez le vétérinaire) et leur fréquence (quotidienne, hebdomadaire, mensuelle, ponctuelle)

Cas 1 et 2 - terminaison :

- Le système enregistre les données et affiche un message de confirmation

### **Suppression de compte**

- Affichage de la fenêtre des paramètres de l'utilisateur
- L'utilisateur sélectionne *Supprimer le compte* dans la danger zone
- Le système supprime toutes les données de l'utilisateur, et le retire de toutes les cartes où il était désigné en tant que personne assignée à la tâche

### **2) Gestion des offres:**

Le système de prestations est basé sur une acceptation mutuelle de l'offre par le prestataire et par le bénéficiaire.

### **Acceptation d'une offre**

Pour le prestataire :

- Affichage de la liste des besoins des bénéficiaires
- Le prestataire sélectionne une offre et propose son aide
- L'application met à jour l'offre avec un statut *En attente de confirmation* et envoie un email au bénéficiaire pour l'informer du changement

Pour le bénéficiaire

- Affichage de la page de l'offre en attente de confirmation, avec les informations des différents prestataires ayant proposé leur aide
- Le bénéficiaire accepte ou refuse l'aide du prestataire
- Le système enregistre le choix du bénéficiaire

Terminaison 1 : Le bénéficiaire accepte la proposition

- Le système envoie un email au prestataire pour l'informer de l'acceptation
- Le système met à jour les calendriers du prestataire et du bénéficiaire

Terminaison 2 : Le bénéficiaire refuse la proposition

- Le système envoie un email au prestataire pour l'informer du refus

### **Suspension d'une offre**

- Affichage de la page des offres du bénéficiaire
- Le bénéficiaire sélectionne l'offre qu'il souhaite suspendre
- Le système affiche les détails de l'offre
- Le bénéficiaire suspend l'offre
- Le système met à jour l'offre en base de données

### **Suppression d'une offre**

- Affichage de la page des offres du bénéficiaire
- Le bénéficiaire sélectionne l'offre qu'il souhaite suspendre
- Le système affiche les détails de l'offre
- Le bénéficiaire suspend l'offre
- Le système supprime l'offre en base de données

### **3) Gestion des prestations:**

### **Annulation d'une prestation**

- Affichage de la page d'une prestation en attente
- Le prestataire renseigne un motif d'annulation

Cas 1 : Prestation ponctuelle

- Le système envoie un email au bénéficiaire l'informant de l'annulation

Cas 2 : Prestation récurrente

- Le prestataire spécifie s'il annule une seule prestation ou s'il annule toutes les prestations à venir

Cas 2.1 : Annulation d'une seule prestation

- Le système envoie un email au bénéficiaire l'informant de l'annulation

Cas 2.2 : Annulation de toutes les prestations (résiliation d'un accord de prestations récurrentes)

- Voir la section *Résiliation d'un accord*

### **Résiliation d'un accord**

Par "accord", on définit une prestation récurrente acceptée par le bénéficiaire et le prestataire. Étant donné qu'un accord engage le prestataire pour une prestation récurrente, une date de préavis de résiliation est à prévoir. Cette durée est définie au moment de l'accord, dans l'*Acceptation de l'offre*

- Affichage de la page d'une offre
- Le prestataire sélectionne *Résilier l'accord* et la date de résiliation de ses services au bénéficiaire
- Le système demande une confirmation et avertit des conséquences en cas de préavis non-respecté
- Le prestataire confirme
- Le système enregistre l'annulation du service et, en fonction de la durée du préavis, applique une amende sur la

rémunération du prestataire

### Évaluation d'une prestation

Prérequis :

- À la fin d'une prestation, le système envoie un email au bénéficiaire et au prestataire contenant un lien vers un formulaire d'évaluation

- Affichage de la page contenant le formulaire d'évaluation
- L'utilisateur et le bénéficiaire saisissent leur évaluation sur la prestation de la journée (accueil, disponibilité, ponctualité, réalisation de la demande)
- Le système enregistre l'évaluation en base de données et recalcule les scores de chaque utilisateur
- Lorsque le prestataire et le bénéficiaire ont chacun évalué, le système rend les résultats de l'évaluation disponible aux deux

### Rémunération d'une prestation

La rémunération des prestations est conditionnée par le type de prestation. Une prestation ponctuelle est rémunérée au moment de la prestation. Pour cela, le bénéficiaire de la prestation dispose d'un délai d'une semaine pour payer la prestation. Une prestation récurrente est rémunérée mensuellement, et un malus est appliqué en cas de non exécution de la prestation.

- Affichage de la page d'une prestation récurrente en cours ou d'une prestation ponctuelle exécutée
- L'utilisateur sélectionne *Rémunérer la prestation*
- Le système redirige l'utilisateur vers la page de paiement pré-remplie avec les informations du paiement de la prestation
- Le bénéficiaire entre ses données de paiement
- Le système vérifie et valide le paiement avec le système tiers de paiement en ligne

### Téléchargement d'une facture

- Affichage d'une prestation qui a été rémunérée
- L'utilisateur choisit de télécharger la facture
- Le système génère et télécharge la facture sur le poste de l'utilisateur

## B. Interface administrateur / TouffuAdmin

...

### C. Pilotage du projet / Touffu Management

La gestion du projet Touffu se fait au travers d'une application Java avec une interface graphique similaire à celle de Trello, comme le montre la figure 6. La vue principale d'un projet se divise en colonnes, dans lesquelles des cartes affichent les différentes tâches.

Les fonctionnalités suivantes sont décrites sous la forme graphique, mais une version de l'application est également disponible en ligne de commande, auquel cas les traitements

sont les mêmes, à l'exception de la gestion des fenêtres d'affichage.

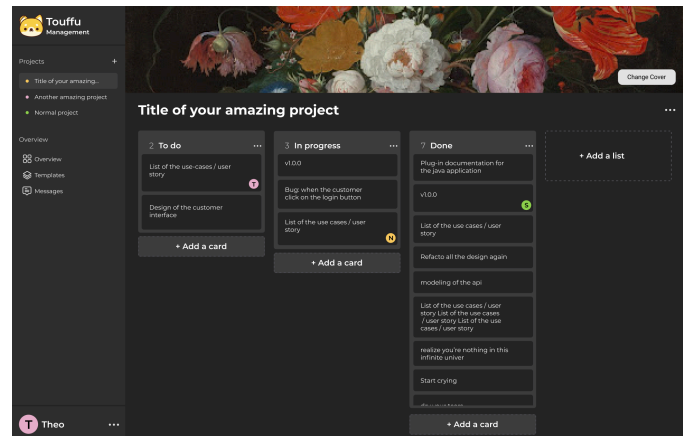


Fig. 6. Management app - home design

#### 1) Gestion des utilisateurs:

##### Inscription

- Affichage d'une fenêtre d'inscription
- L'utilisateur renseigne son email (doit être unique dans l'instance de la base de données), un mot de passe et son nom complet (nom, prénom)
- L'application vérifie l'unicité de l'email et insère les données en base de données
- L'application envoie un email de confirmation d'inscription à l'utilisateur
- L'utilisateur est redirigé vers la fenêtre de connexion

##### Connexion

- Affichage d'une fenêtre de connexion
- L'utilisateur entre ses identifiants de connexion (email et mot de passe)
- L'application vérifie la combinaison des identifiants
- En cas d'erreur, la fenêtre affiche un message d'erreur
- En cas de succès, l'utilisateur est redirigé vers la page d'accueil

##### Suppression de compte

- Affichage de la fenêtre des paramètres de l'utilisateur
- L'utilisateur sélectionne *Supprimer le compte* dans la danger zone
- Le système supprime toutes les données de l'utilisateur, et le retire de toutes les cartes où il était désigné en tant que personne assignée à la tâche

#### 2) Gestion des projets et des tâches:

##### Création d'un projet

- Affichage d'une fenêtre pour créer un nouveau projet
- L'utilisateur entre le nom du projet, (**la deadline et ?**) choisit une image de fond
- Le système insère le projet en base de données

- Le système enregistre l'image dans les données de l'application (...-¿ Serveur, bdd, local...?)

#### **Modification d'un projet**

- Affichage de la fenêtre de modification du projet
- L'utilisateur modifie le champ du projet qu'il souhaite modifier (nom, couleur, etc.).
- Le système enregistre les changements

#### **Ajout d'un utilisateur à un projet**

- Affichage de la fenêtre de modification du projet
- Dans la section *Membres*, l'utilisateur clique sur le bouton *Ajouter un membre*
- L'utilisateur entre l'email de la personne
- L'application envoie un email pour informer l'utilisateur qu'il a été ajouté dans le projet

#### **Suppression d'un projet**

- Affichage de la fenêtre de modification du projet
- L'utilisateur clique sur le champ *Supprimer le projet* dans la danger zone
- Le système supprime le projet et toutes les cartes, listes et tâches associées en base de données

#### **Ajout d'une liste**

- Affichage d'une fenêtre de création de liste
- L'utilisateur entre le nom de la nouvelle liste
- L'application ajoute la liste en base de données et l'affiche sur le tableau du projet

#### **Modification d'une liste**

- Affichage de la fenêtre de modification de la liste
- L'utilisateur modifie le champ de la liste qu'il souhaite modifier (nom).
- Le système enregistre les changements

#### **Suppression d'une liste**

- Affichage de la fenêtre de modification de la liste
- L'utilisateur clique sur le champ *Supprimer la liste* dans la danger zone
- Le système supprime la liste ainsi que les tâches qu'elle contenait

#### **Ajout d'une tâche**

- Affichage d'une fenêtre d'ajout d'une carte
- L'utilisateur entre le nom de la carte, ainsi qu'un utilisateur assigné à cette tâche (facultatif)
- L'application ajoute la tâche en base de données et affiche la carte dans la liste associée

#### **Modification d'une tâche**

- Affichage de la fenêtre de modification de la tâche
- L'utilisateur modifie le champ de la tâche qu'il souhaite modifier (nom, utilisateur associé, liste).
- Le système enregistre les changements

#### **Suppression d'une tâche**

- Affichage de la fenêtre de modification de la tâche
- L'utilisateur clique sur le champ *Supprimer la carte* dans la danger zone
- Le système supprime la carte

#### **Export PDF**

- To do

#### *3) Gestion des plugins:*

#### *4) Messagerie interne:*

La messagerie d'un utilisateur lui permet d'échanger avec les différents membres de son équipe. Les messages se font entre deux utilisateurs donnés.

#### **Affichage d'une conversation**

- L'utilisateur clique sur l'icône d'un autre utilisateur dans une carte
- Le système ouvre la messagerie sur la conversation avec cet utilisateur

OU

- L'utilisateur ouvre la messagerie
- Le système ouvre la page par défaut de la messagerie
- L'utilisateur sélectionne une conversation déjà entamée dans la liste de ses conversations

#### **Lancement d'une conversation**

- Affichage de la fenêtre par défaut de la messagerie
- L'utilisateur clique sur le bouton + (*Nouvelle conversation*)
- Le système affiche un champ de recherche
- L'utilisateur entre le nom, le prénom ou l'email de l'utilisateur avec qui il souhaite initier une conversation
- Le système recherche dans la liste des utilisateurs de l'application une correspondance
- Si aucune correspondance n'est trouvée, la fenêtre affiche un message d'erreur
- Si une ou plusieurs correspondances sont trouvées, l'utilisateur sélectionne le bon utilisateur
- L'application ouvre une fenêtre d'envois de messages

#### **Envoi d'un message**

Prérequis : Affichage d'une conversation

- L'utilisateur envoie un message par le champ d'envoi
- Le système enregistre le message

#### *D. Récupération de données (Scraping)*

...

### **III. ENVIRONNEMENT DE DÉVELOPPEMENT**

#### *A. Décisions logicielles*

...

#### *B. Estimation des coûts*

...

## IV. SPÉCIFICATIONS

...

### V. ARCHITECTURE LOGICIELLE

#### A. Architecture globale

...

#### B. Organisation des répertoires

Le projet est développé et versionné sous Git, dans une organisation appelée touffu-esgi<sup>[3]</sup>, et administré par les développeurs. L'organisation comporte six "repos" majeurs, qui sont les dossiers des modules de l'application. Ces repos sont indépendants les uns des autres en termes de code propre, mais interdépendants en ce qui concerne la donnée, tel que le montre la figure 1.

*Touffu*<sup>[4]</sup> : Le répertoire Touffu contient le front-end principal, soit l'application client en AngularJS.

*TouffAdmin*<sup>[5]</sup> : TouffAdmin est le repository du backend administratif de l'application Touffu, lui aussi en AngularJS

*TouffApi*<sup>[7]</sup> : Le répertoire TouffApi contient les éléments de l'API en NodeJS.

*Touffu-Management*<sup>[6]</sup> : Ce répertoire est dédié à l'application lourde en Java.

*Touffu-noSQL*<sup>[9]</sup> : Le répertoire Touffu-noSQL contient toutes les données relatives à la base de données NoSQL.

*Touffu-Scrap*<sup>[8]</sup> : Touffu-Scrap est le répertoire destiné au langage de scraping développé par touffu-esgi afin de récupérer des données sur Internet au sujet des services à la personne.

#### C. Architecture des modules

*Touffu:*

...

*TouffAdmin:*

...

*TouffApi:*

...

*Touffu-Management:*

...

*Touffu-noSQL:*

...

*Touffu-Scrap:*

...

## VI. IMPLÉMENTATION

Cas d'utilisation...

### REFERENCES

- [1] <https://www.servicesalapersonne.gouv.fr/beneficier-des-sap/queles-sont-activites-de-services-la-personne/soins-et-promenades-d-animaux-de-compagnie-pour-personnes-dependantes>
- [2] <https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000000789087>
- [3] <https://github.com/touffu-esgi>
- [4] <https://github.com/touffu-esgi/Touffu>
- [5] <https://github.com/touffu-esgi/TouffAdmin>
- [6] <https://github.com/touffu-esgi/Touffu-Management>
- [7] <https://github.com/touffu-esgi/TouffApi>
- [8] <https://github.com/touffu-esgi/Touffu-Scrap>
- [9] <https://github.com/touffu-esgi/Touffu-noSQL>