



## DIVA'NUM

Création d'un carnet de bord numérique pour le recueil d'expérience des aidants pour DIVADOM

**Livrable finale**

Groupe 13: Mamisoa RANDRIANARIMANANA- Chenrui ZHU - Xianxiang ZHANG  
Tuteurs Polytech: Yannick PRIÉ - Jean-Philippe RIVIÈRE  
Tutrice entreprise: Alice LAGRÈVE

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Presentation du projet . . . . .	2
1.2	Présentation de l'équipe . . . . .	2
<b>2</b>	<b>Solution proposée</b>	<b>4</b>
2.1	Utilisateur . . . . .	4
2.2	Description de DIVA'NUM . . . . .	4
2.3	Cas d'utilisation et scénario . . . . .	5
<b>3</b>	<b>Démonstration de l'application Web en l'état et cas de test</b>	<b>9</b>
<b>4</b>	<b>Architecture de l'application Web DIVANUM</b>	<b>17</b>
4.1	La couche de présentation (Frontend) . . . . .	17
4.2	La couche applicative (Backend) . . . . .	18
4.3	La couche de persistance des données . . . . .	20
4.4	API . . . . .	21
<b>5</b>	<b>Gestion de projet</b>	<b>23</b>
<b>6</b>	<b>Bilan entre ce qui a été demandé et ce qui a été fait</b>	<b>26</b>
<b>7</b>	<b>Conclusion</b>	<b>28</b>
<b>8</b>	<b>Annexe</b>	<b>29</b>
8.1	Contexte technologique . . . . .	29

# 1 Introduction

## 1.1 Présentation du projet

Pour répondre au défi du vieillissement de la population, la Ville de Nantes, avec ses partenaires institutionnels et associatifs, soutient des politiques publiques et dispositifs dédiés à l'accompagnement de la transition démographique notamment par un souci d'amélioration du service rendu aux seniors nantais et de préparation de la transition démographique. DIVADOM ou Dispositif Innovant de Vie à domicile en fait partie.

DIVADOM a pour objectif d'un côté proposer un accompagnement renforcé 24h/24 à des personnes âgées (plus de 60 ans) en perte d'autonomie souhaitant rester à domicile comme une alternative à l'entrée en EHPAD. De l'autre côté, le dispositif a également pour but d'apporter un soutien aux aidants qui sont des proches du bénéficiaire de DIVADOM et qui a pour rôle de veiller et d'aider la personne âgée dans son quotidien.

Afin d'évaluer le dispositif, une sonde culturelle orientée vers les aidants a été mise en place, amenant à la naissance de ce projet transversal avec les étudiants en 4ème année d'Informatique de Polytech Nantes, où le porteur de projet qui est DIVADOM souhaite avoir une plateforme numérique de recueil d'expérience de l'aidant qui viendra compléter un carnet de bord physique.

Cette plateforme qu'on a nommée DIVA'NUM permettra aux aidants des personnes âgées d'envoyer et de sauvegarder des données numériques ainsi que de répondre à des questionnaires qui seront ensuite recueillies et analysées pour évaluer le dispositif.

Ce projet couvre la spécification des besoins, la conception de DIVA'NUM jusqu'à sa production en une plateforme utilisable répondant aux besoins des utilisateurs spécifiés dans le cahier des charges. Le projet durera 8 mois de Septembre à Mai. La méthode de gestion de projet imposée est la méthode agile. Ainsi, le projet sera décomposé en plusieurs itérations de 4 semaines.

## 1.2 Présentation de l'équipe

Pour le projet, la représentante du client sera Alice LAGREVE, qui est une coordinatrice de DIVADOM. Elle joue le rôle de MOA en représentant et en exprimant les besoins de l'entreprise et des utilisateurs finaux. Dans la même vocation, il y a également Jean-Philippe RIVIÈRE et Yannick PRIE, tous les deux enseignants au sein de l'école POLYTECH NANTES, qui sont les encadrants du projet. Ils accompagnent les étudiants et l'entreprise dans le bon déroulement du projet, jouant le rôle à la fois de MOA et MOE, c'est-à-dire qu'ils prennent part tant dans l'aspect fonctionnel et technique du projet.

De l'autre côté, en MOE, il y aura les étudiants: Mamisoa RANDRIANARIMANANA, Chenrui ZHU et Xianxiang ZHANG qui sont tous les trois étudiants en quatrième année en Informatique à Polytech Nantes. Ils seront chargés de la création et du développement de DIVA'NUM. Mamisoa aura le rôle de chef de projet, Chenrui de responsable BackEnd et Xianxiang de responsable FrontEnd.

Dans le projet, on aura alors six acteurs décrits dans le tableau ci-dessous.

Nom	Rôle	Contact
Alice LAGREVE	MOA Représentante DIVADOM	<a href="mailto:alice.lagreve@mairie-nantes.fr">alice.lagreve@mairie-nantes.fr</a>
Yannick PRIE	MOA Encadrant POLYTECH	<a href="mailto:yannick.prie@univ-nantes.fr">yannick.prie@univ-nantes.fr</a>
Jean-Philippe RIVIERE	MOA Encadrant POLYTECH	<a href="mailto:jp-riviere@univ-nantes.fr">jp-riviere@univ-nantes.fr</a>

Mamisoa RANDRIANARIMANANA	MOE Chef de projet	<a href="mailto:mamisoa.randrianarimanana@etu.univ-nantes.fr">mamisoa.randrianarimanana@etu.univ-nantes.fr</a>
Chenrui ZHU	MOE Responsable BackEnd	<a href="mailto:chenrui.zhu@etu.univ-nantes.fr">chenrui.zhu@etu.univ-nantes.fr</a>
Xianxiang ZHANG	MOE Responsable FrontEnd	<a href="mailto:xianxiang.zhang@etu.univ-nantes.fr">xianxiang.zhang@etu.univ-nantes.fr</a>

## 2 Solution proposée

Afin de répondre aux besoins des porteurs de projet dont la nature est, en premier lieu, de lier le carnet de bord physique et numérique afin que les aidants puissent envoyer des ressources numériques en complémentaires des retours d'expérience écrits dans le carnet de bord physique, on a proposé comme solution une plateforme numérique que nous avons nommé DIVA'NUM qui est une application Web.

À ce besoin s'ajoute la mise en place de questionnaire que l'évaluateur pourra créer que l'aidant pourra remplir.

Enfin, les évaluateurs de DIVADOM devront être capable de recueillir tous les données ajoutées et complétées par les aidants.

Le lien entre les deux carnets qui va être mis en place se fera de deux manières:

- un lien par connexion grâce à un identifiant et un mot de passe qui sera fourni dans le carnet et permettra à l'aidant de faire plusieurs actions ;
- un lien par un code que l'aidant entrera sur un plateforme direct et qui lui permettra d'ajouter un fichier numérique ou répondre directement à un questionnaire; Ces codes seront inscrit dans le carnet physique lors d'une invitation à envoyer un fichier ou remplir un questionnaire. Cette deuxième solution a été mise en place pour faciliter aux maximum l'ajout de fichier et la réponse aux questionnaires sans obliger l'aidant à se connecter.

On détaillera par la suite les différents actions que pourront faire chaque utilisateur via le plateforme.

### 2.1 Utilisateur

DIVANUM aura deux utilisateurs: d'un côté l'évaluateur et de l'autre côté l'aidant.

1. **Évaluateur:** Ce sont les chercheurs liés au DIVADOM qui recueillent les données fournies par les aidants dans ce système et les utilisent comme référence pour de future évaluation de DIVADOM.
2. **Aidant:** D'après la définition de l'aidant extraite de la loi n°2015-1776 du 28 décembre 2015 (article 51) le proche aidant est défini comme étant une "personne qui vient en aide, de manière régulière et fréquente à titre non professionnel, pour accomplir tout ou partie des actes ou des activités de la vie quotidienne d'une personne en perte d'autonomie, du fait de l'âge, de la maladie ou d'un handicap". Ce sont ainsi les personnes qui aident les personnes âgées pour pouvoir les maintenir à domicile. Dans notre projet, c'est un utilisateur qui va exprimer son vécu, son expérience en complétant le carnet de bord physique et en complétant avec des données numériques qu'il mettra dans le carnet de bord numérique tel que des photos, mémo vocaux... c'est lui qui fournira les données qui seront évalués par les évaluateurs ensuite. Ce sont tous des gens ordinaires, principalement des enfants ou des compagnons de personnes âgées, très âgés et évidemment moins rapides que les jeunes dans l'utilisation des smartphones ou des tablettes. Pour eux, comment utiliser le système pour se connecter et saisir les données rapidement et facilement est la chose la plus importante.

### 2.2 Description de DIVA'NUM

DIVA'NUM est une application Web conçu pour deux utilisateurs différents: D'un côté le proche aidant qui pourra envoyer des données numériques. De l'autre côté l'évaluateur qui pourra récupérer les fichiers des aidants.

Chaque utilisateur aura une interface différente suite à la connexion. Pour accéder à la page de connexion, on clique sur le bouton "Connexion" et on arrive sur la page de connexion où on doit renseigner un identifiant et un mot de passe qui seront vérifiés par le système. Dès que l'authentification s'est bien déroulé, le système affecte automatiquement l'interface correspondant à chaque usager selon leur rôle soit AIDANT soit EVALUATEUR.

Afin de faciliter l'envoi de fichier numérique, une fonctionnalité d'ajout de fichier via une accès rapide ne nécessitant pas d'authentification a été mise en place. L'aidant n'aura qu'à entrer un code présent dans son carnet pour être reconnu par le système et envoyer ses fichiers numériques. Il pourra également voir les fichiers qu'il a déjà envoyés auparavant. Notons que ce code ne pourra être utilisé qu'une seule fois dès que l'ajout a été validé.

L'accès à cette plateforme directe se fait via le bouton "Accès rapide".

Pour l'instant, l'interface aidant ne lui permet que d'ajouter des fichiers numériques, de voir ses contenus ainsi que de récupérer ses informations notamment la liste des codes pour ajouter les fichiers via l'accès rapide. Mais on l'a conçu pour permettre des évolutions notamment la modification ou suppression de ses ajouts, la réception de notification venant des évaluateurs ou interactions par messageries, etc.

Afin de faciliter la compréhension, nous allons décrire les différents actions des utilisateurs sous forme de scénarios d'utilisation.

## 2.3 Cas d'utilisation et scénario

On va décrire les parcours utilisateurs avec des scénarios à l'aide de persona. Un Persona est une représentation fictive de l'utilisateur cible. Elle permet de définir les besoins, les problématiques et les questions que se posent les utilisateurs potentiels tout au long de son parcours lors de l'utilisation de la plateforme.

### Scénario 1: Envoie d'un fichier numérique

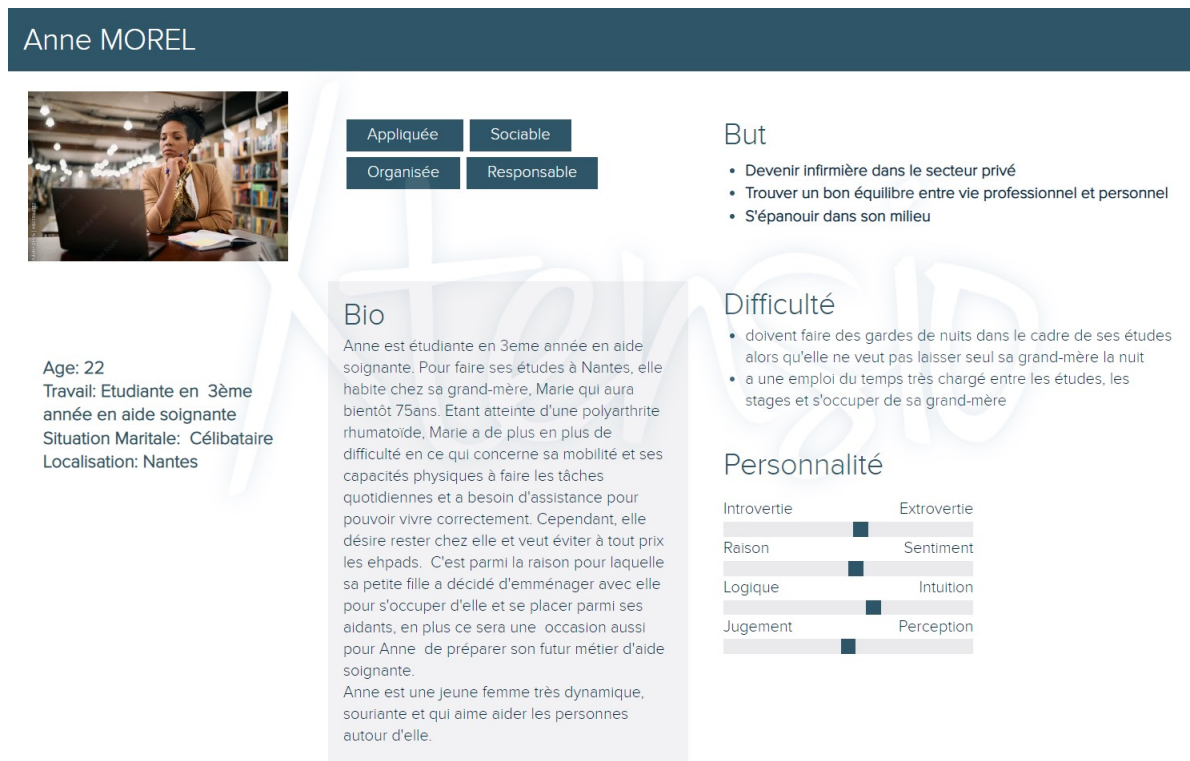


Figure 1: Profil de Anne, proche aidant

Afin de pouvoir soulager la maladie de Marie, son médecin lui a conseillé la balnéothérapie. En regardant les différentes possibilités, Anne va alors amener sa grand-mère au centre balnéothérapie qui est situé à 45mn de leur domicile. N'ayant pas de véhicule, Anne a pu, grâce à DIVADOM bénéficier d'un transport adapté pour pouvoir amener sa grand-mère. Le soir, après leur sortie, Anne reçoit une notification sur la tablette dédiée au DIVADOM chez Anne, l'invitant à partager son expérience sur le service de transport recommandé par DIVADOM. Elle prend alors le carnet de bord physique, et complète le questionnaire relatif aux services de transport. Dans son carnet, elle a une invitation pour envoyer des fichiers numériques, elle aimerait faire un témoignage audio. Elle enregistre alors son audio avec sa tablette. Elle est prête à l'envoyer. Elle va sur le plateforme direct de DIVA'NUM, entre le code T45XR3 et envoie son fichier.

## Scénario 2: Répondre à un questionnaire



Figure 2: Profil de Simone, proche aidant

Afin de mieux gérer les différents transferts de ses parents, qui ne peuvent plus se déplacer et se mettre de debout du fauteuil au lit ou du fauteuil vers la salle de bain, Simone a prévu de mettre en place un système de lève personne pour éviter de les porter car elle a aussi des problèmes de dos et n'y arrive plus. Elle s'est tournée alors vers Divadom pour lui conseiller mais également d'apprendre à l'utiliser. Au bout d'une semaine, comme l'avait demandé l'équipe de Divadom, elle a décidé de remplir l'expérience relative à la session d'information et de formation. N'ayant pas l'habitude avec la technologie, Simone appréhende beaucoup le fait de devoir aller sur Diva'Num. Heureusement, que dans toutes les tablettes du Divadom, l'application Web est directement mise en favori. Elle n'a qu'à cliquer dessus et elle arrive directement sur l'accès direct. Elle arrive sur l'accès direct de DIVA'NUM, entre le code Q5GD25 et arrive sur un questionnaire. Elle y répond puis clique sur terminer.

### Scénario 3: Récupérer les ajouts d'un aidant



Figure 3: Profil de Hugo, évaluateur de DIVADOM

Hugo est un coordinateur de DIVADOM. Il joue également le rôle d'évaluateur dans le dispositif. Pour avoir une vue générale de l'expérience des aidants sur un service, il doit récupérer les expériences des aidants, de les classer et de les analyser puis les synthétiser pour donner un avis d'amélioration dessus. Etant donné que c'est la fin de l'expérience pour l'aidant 2, il a récupéré son carnet et il doit maintenant récupérer ses ajouts numériques sur DIVA'NUM. Il va sur le plateforme de connexion de DIVA'NUM, entre son identifiant et son mot de passe, puis arrive sur la page d'accueil. Il clique sur "Aidant 2" et peut voir tous ses contenus. Il clique ensuite sur "Télécharger". Il obtient alors sur son ordinateur un dossier sous format zip au nom de l'Aidant 2 contenant tous ses ajouts. Il peut maintenant procéder à son évaluation.

### Scénario 4 : Créer un questionnaire

Hugo doit créer des questionnaires pour connaître les attentes des aidants sur le dispositif. Il a une liste de questions qui lui a été fournie. Il doit alors le créer pour que les aidants puissent y répondre dans la semaine qui suit. Il se connecte alors en tant qu'évaluateur sur DIVA'NUM. Il clique sur "Créer Questionnaire". Il arrive sur une page lui permettant d'ajouter des questions, de proposer plusieurs choix de réponses ou laisser un champ de réponse libre. Il rédige alors ses questions. Quand il a fini, il clique sur "Terminer". Son questionnaire est alors créé. Les aidants pourront y répondre en entrant des codes spécifiques créés uniquement pour ce questionnaire et qui est disponible dans la liste des codes de chaque aidant visible dans la partie "Gestion des aidants".



Pour avoir une visualisation plus détaillée avec les différentes étapes, on a modélisé les différents actions et les points d'entrées via les figures 4 et 5 suivant:

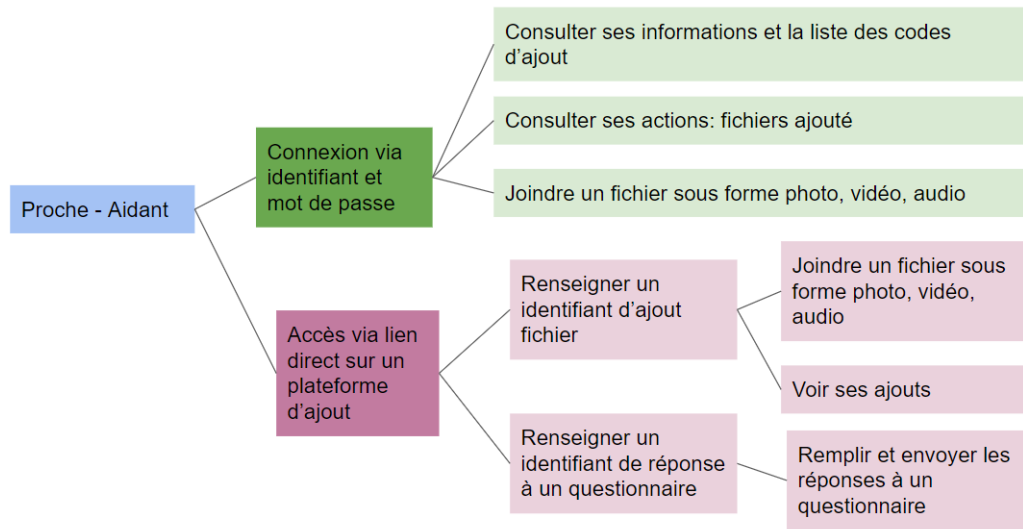


Figure 4: Point d'entrée et activité possible de l'Aidant

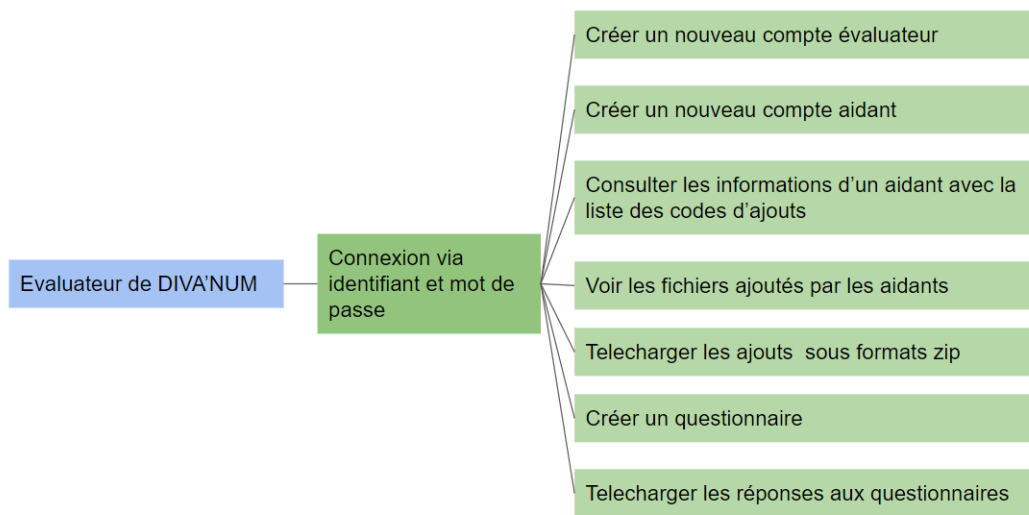


Figure 5: Point d'entrée et activité possible de l'Évaluateur

### 3 Démonstration de l'application Web en l'état et cas de test

#### Accueil:

L'image 6 ci-dessous est l'interface principale de divanum. Différentes opérations peuvent être effectuées via le bouton de connexion et d'accès rapide.



Figure 6: Ecran d'accueil

#### Connexion:

- Authentification en tant qu'aidant ou évaluateur.
- Traite le cas d'un identifiant ou mot de passe invalide.
- Charge la page d'accueil aidant si aidant.
- Charge la page d'accueil évaluateur si évaluateur.

Table 2 permet de vérifier les tests réalisés.

Table 2: Test Connexion

Cas de test	Résultat attendu	Résultat obtenu
Connexion avec identifiant et mot de passe aidant.	Accès à la page d'accueil de l'aidant.	OK
Connexion avec identifiant et mot de passe évaluateur	Accès à la page d'accueil de l'évaluateur	OK
Connexion avec identifiant ou mot de passe manquant ou erroné	Connexion impossible, message d'erreur	OK

L'interface de connexion Figure 7 est simple, avec des champs de saisie d'identifiant et de mot de passe présentés.

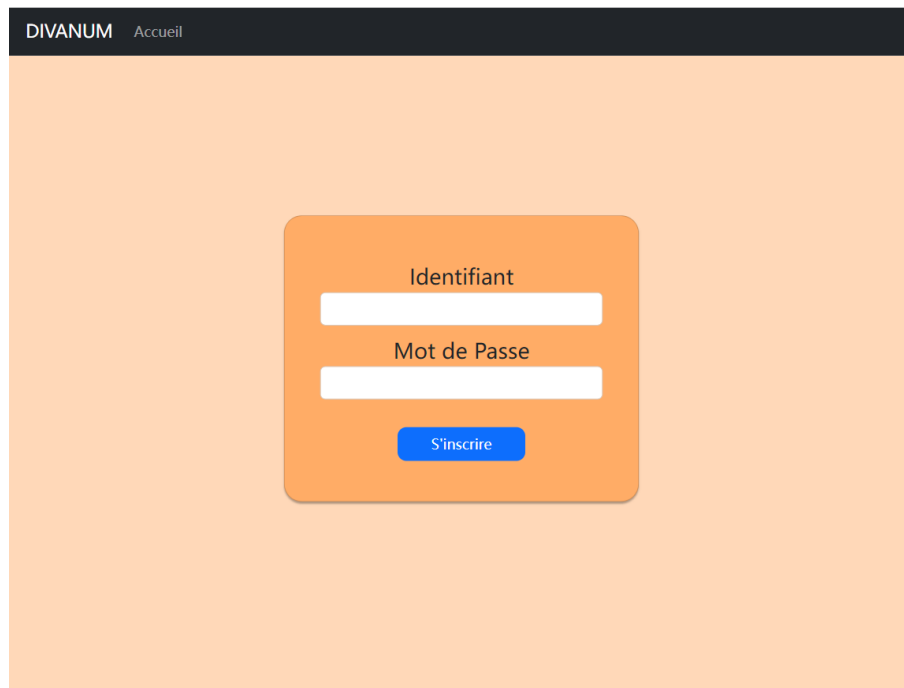


Figure 7: Interface de connexion

### Plateforme direct:

L'utilisateur peut passer à l'étape suivante en entrant le code.<sup>8</sup> Le système peut dire par lui-même de quel type de code il s'agit et s'il est valide.

Figure 9 s'agit de l'interface après la saisie du code du fichier, et l'utilisateur peut ajouter des fichiers en cliquant sur le bouton au milieu de l'écran.

Figure 10 s'agit de l'interface après la saisie du code du questionnaire et l'utilisateur effectue les tâches associées en remplissant le questionnaire et en cliquant sur le bouton Terminer.

- reconnaissance du code
- envoie du fichier
- le fichier est sauvegarder comme un fichier de l'aidant
- utilisation d'une code une seule fois, dès que c'est validé, on ne peut plus le réutiliser
- la taille du fichier ne dépasse pas 200Mo
- message de confirmation après validation que tout s'est bien passé
- message d'erreur si le code n'existe pas
- après que le code soit entré, l'aidant peut avoir un aperçu de tous les fichiers qu'il a envoyé

Table 3 présente des scénarios de test spécifiques pour l'interface de connexion rapide.

Table 3: Test plate-forme direct

Cas de test	Résultat attendu	Résultat obtenu
-------------	------------------	-----------------

Connexion à la plateforme directe		
On entre un code d'ajout de fichier valide d'un aidant	Accès à la page d'ajout direct de fichier + aperçu des fichiers ajoutés ultérieurement de l'aidant concerné	OK
On entre un code de questionnaire valide d'un aidant	Accès au questionnaire et possibilité de répondre et validé. La réponse devra être visible dans "Voir réponse aux questionnaires" par l'évaluateur	OK
On entre un code n'appartenant à aucun aidant	Message d'erreur	OK
On entre un code qui a déjà utilisé par l'aidant	Message d'erreur	OK
Ajout fichier via plateforme directe		
Ajout d'un ou plusieurs fichiers de n'importe quel type (mp3, mp4, jpeg, etc) ne dépassant pas 200Mo	Message de succès + Fichier ajouté et visite sur la liste des fichiers de l'aidant	OK
Ajout d'un fichier dépassant 200Mo	Message d'erreur	OK

### Page d'accueil Aidant:

Figure 11 s'agit de l'interface après la connexion de l'utilisateur. Étant donné que les principales fonctions sont placées sur la plate-forme rapide, le but de cette connexion est uniquement de visualiser les codes et de changer le mot de passe, et de fournir la possibilité d'améliorations futures.

- vérifier les codes
- changer le mot de passe

Table 4 présente des scénarios de test spécifiques pour l'interface de l'aidant.

Table 4: Test Page Aidant

Cas de test	Résultat attendu	Résultat obtenu
Affichage des codes		
Affichage des codes des documents et des questionnaires séparément	Affichage sans erreur	OK
Modification du mot de passe	Message d'erreur	OK

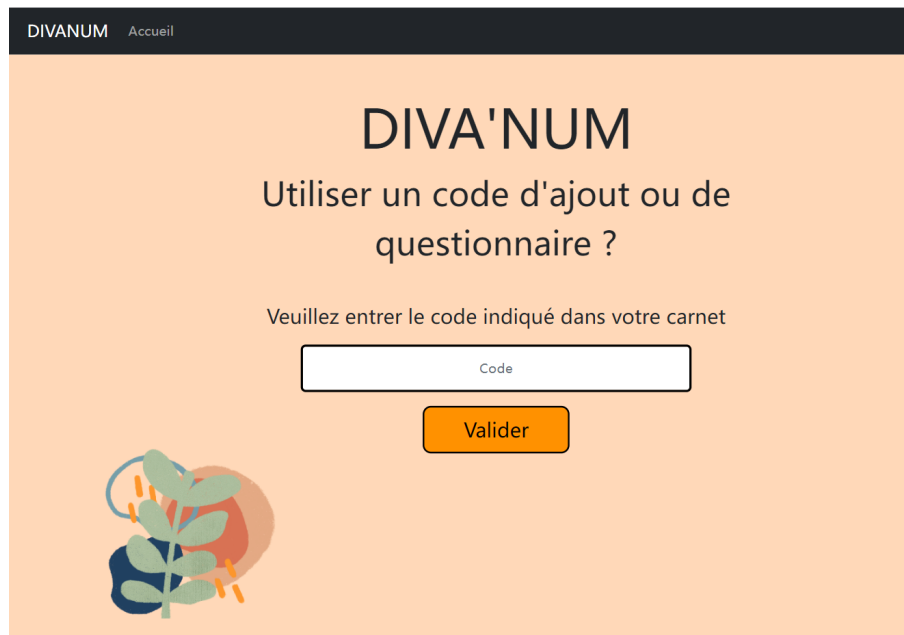


Figure 8: Interface de plateforme direct

### Evaluateur:

Figure 12 s'agit de l'interface après la connexion de l'administrateur, qui implique l'affichage des utilisateurs, le téléchargement des fichiers d'utilisateurs et des boutons tels que la création de questionnaires et la création d'utilisateurs.

L'image 13 est l'interface à laquelle l'administrateur accède après avoir cliqué pour créer un questionnaire. À l'aide de l'API surveyjs, l'administrateur peut directement effectuer diverses opérations dessus et l'enregistrer via l'image Enregistrer dans le coin supérieur droit. Une fois le questionnaire enregistré, le code du questionnaire sera automatiquement généré et envoyé à tous les utilisateurs. Chaque création ultérieure lira le dernier questionnaire enregistré.

La figure 14 est l'interface permettant à l'administrateur de gérer les utilisateurs, et de cliquer sur l'utilisateur pour afficher les codes correspondants.

- enregistrer un nouvel utilisateur → mettre identifiant son mot de passe provisoire  
Si le nouvel utilisateur est un aidant, il apparaît dans la liste des aidants dans gestion aidant avec ses informations: identifiant et les codes pour les ajouts de fichiers
- cliquer sur chaque aidant et voir la liste des fichiers qu'il a ajouter + téléchargement
- créer un questionnaire

Table 5 détaille les tests de l'interface administrateur.

Table 5: Test Page Evaluateur

Cas de test	Résultat attendu	Résultat obtenu
Télécharger les ajouts d'un aidant		

Cliquez sur télécharger contenu d'un aidant	Téléchargement sous format zip d'un dossier au nom de l'identifiant de l'aidant contenant tous ses ajouts	OK
Enregistrer un nouvel utilisateur		
Inscription d'un nouveau aidant	Nouvel aidant visible dans la liste des aidants avec son identifiant et la liste des codes pour le plateforme direct + section pour télécharger ses fichiers + Authentification avec le nouveau identifiant et mot de passe est un succès et est authentifié comme aidant	OK
Inscription d'un nouvel évaluateur	Message de succès + Authentification avec le nouveau identifiant et mot de passe est un succès et est authentifié comme évaluateur	OK
Télécharger les fichiers d'un aidant	Telecharger sous format zip de tous les fichiers correspondant à l'aidant sélectionné	OK
Gérer les aidants	Voir liste des aidants avec son identifiant et la liste des codes pour le plateforme direct + section pour télécharger ses fichiers possibilité d'inscrire un nouveau aidant + possibilité d'inscrire un nouveau aidant	OK
Création d'un questionnaire		
Création d'un questionnaire en cliquant sur créer questionnaire - renseignant les questions et valider	Création d'un nouveau code questionnaire, différent pour chaque aidant, visible dans la liste des codes des aidants et qui renvoie sur le questionnaire quand on entre le code dans le plateforme directe	OK



Figure 9: Accès avec code du fichier

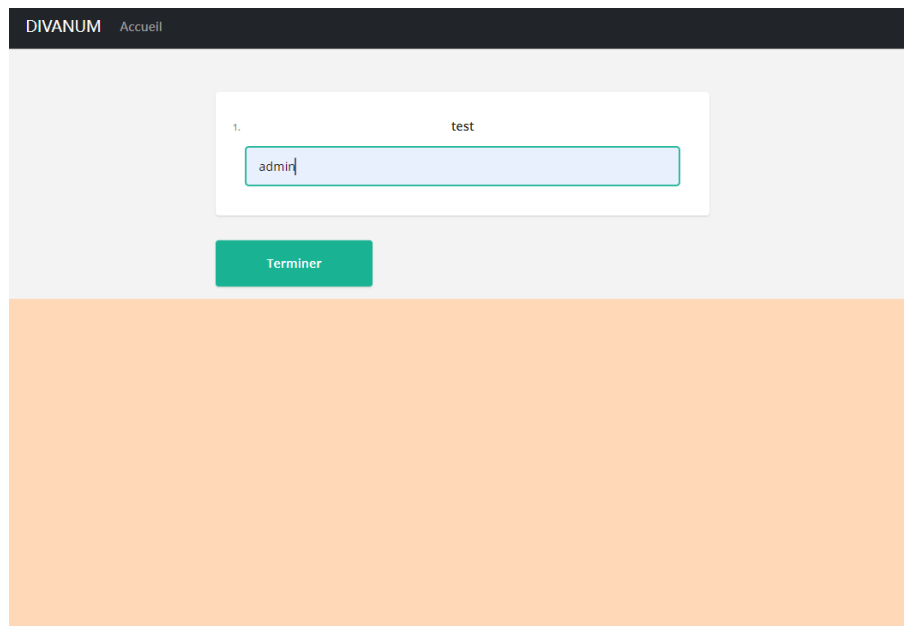


Figure 10: Accès avec code du questionnaire



Figure 11: Profil de l'utilisateur



Figure 12: Interface de l'évaluateur



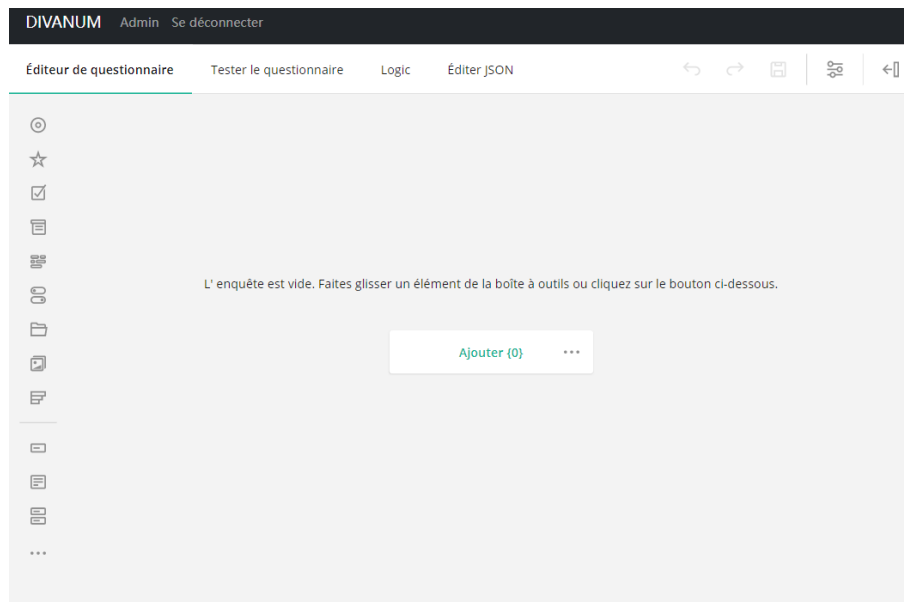


Figure 13: Création de questionnaires



Figure 14: Gestion des aidants

## 4 Architecture de l'application Web DIVANUM

Afin de faire l'application Web, on a choisi d'utiliser des frameworks qui sont un ensemble d'outils, de bibliothèques et de conventions de développement et qui permettent de simplifier et d'accélérer la création d'applications ainsi que de fournir une structure cohérente à l'ensemble de l'application facilitant l'évolutivité et la maintenance du projet.

On a choisi d'utiliser Angular pour le FrontEnd, SpringBoot pour le Backend et MySQL pour la base de données. L'utilisation de ses 3 frameworks permet la séparation claire des préoccupations entre la logique métier (backend), la présentation (frontend) et le stockage de données (base de données).

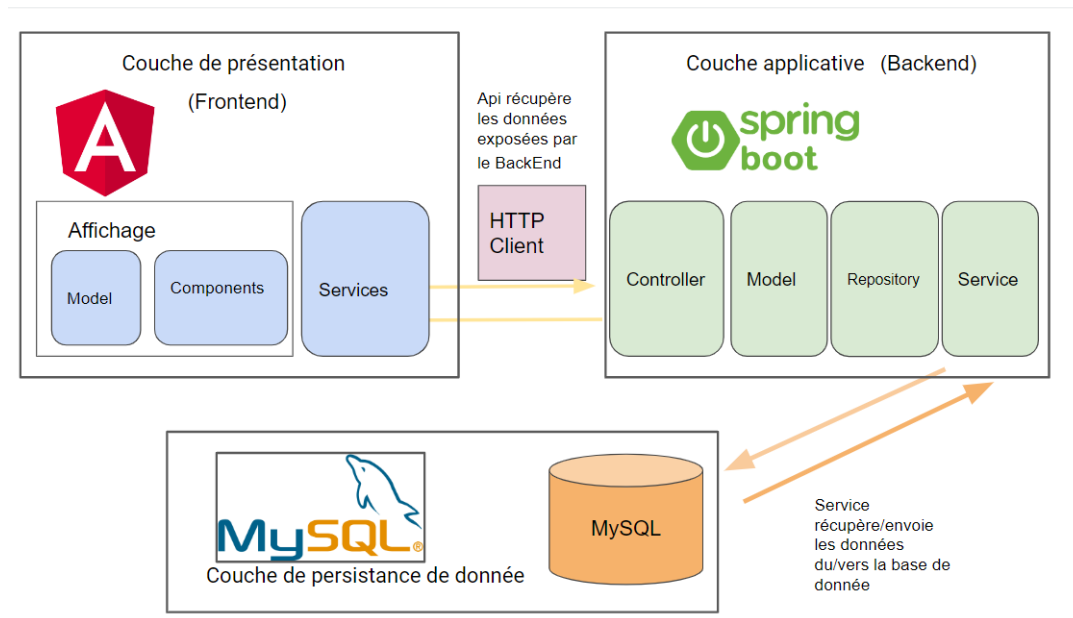


Figure 15: Architecture de l'Application Web DIVANUM

On a alors une architecture en couches :

### 4.1 La couche de présentation (Frontend)

Cette couche est responsable de l'interface utilisateur de l'application et de la gestion des interactions avec l'utilisateur. Elle est constituée de la partie affichage où on sépare les différentes entités en composants et les services.

Les composants (component) qui sont des entités regroupant du code en TypeScript, du HTML et du CSS pour fournir une fonctionnalité ou une partie d'interface utilisateur réutilisable. Elles permettent de diviser une application en éléments plus petits, plus faciles à gérer et à développer, ce qui permet de créer des applications plus modulaires et évolutives. On a mis dans différents composants les différentes fonctionnalités et entités du projet notamment la connexion, le questionnaire, la page d'accueil, la plateforme direct, etc.

Les services sont des classes qui fournissent une fonctionnalité spécifique à l'ensemble de l'application notamment pour le partage des données, les appels de données du Backend comme les appels d'API par exemple le service pour récupérer les fichiers d'un aidant, etc.

Dans notre projet, on aura les principaux composants:

- **board-admin:** qui correspond à la partie de l'évaluateur notamment à l'interface et les différentes actions qu'il pourra faire, il sera relié à d'autres composants comme "caregiver-moderator"
- **board-user:** qui correspond à l'interface de l'aidant notamment à la page d'accueil, et l'ajout de fichier "file-upload"

- **caregiver-moderator:** correspond à la création d'un nouveau compte aidant relié avec le composant "register" ainsi que la liste des aidants existant avec leur informations.
- **components/file-upload:** correspond à l'ajout de fichier pour l'aidant
- **home:** correspond au point d'entrée de l'application Web avec les bouton connexion et plateforme direct
- **login:** correspond à la connexion et l'authentification
- **profile:** permet d'avoir les détails de l'utilisateur : identifiant, rôle, code..
- **quick-access:** correspond au plateforme directe d'ajout de fichier
- **register:** correspond à l'inscription d'un nouvel utilisateur
- **register<sub>admin</sub>:** *correspond à l'inscription d'un nouvel évaluateur* **survey:** *correspond à la gestion des questionnaires*
- **survey-creator:** correspond à la création de nouveau questionnaire

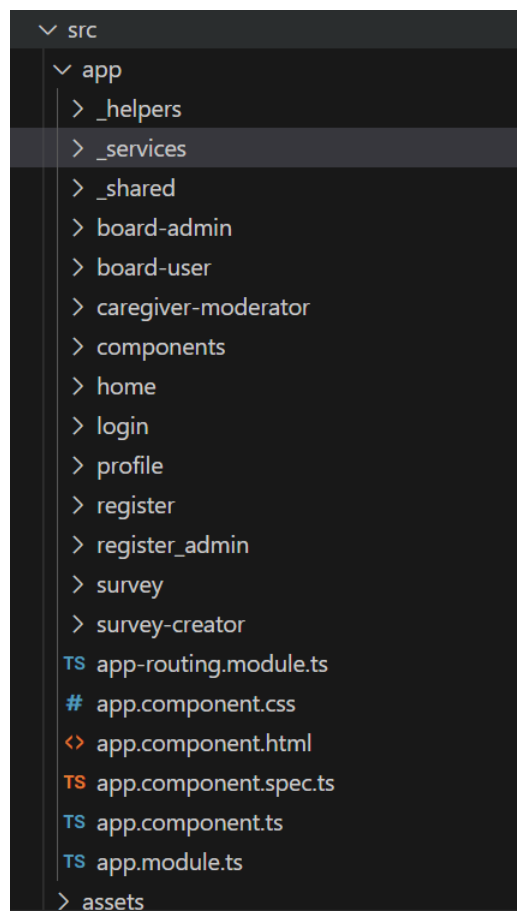


Figure 16: Architecture du frontend

## 4.2 La couche applicative (Backend)

Cette couche est responsable de la logique métier de l'application, de la gestion des requêtes entrantes provenant de la couche de présentation et de l'interaction avec la base de donnée.

L'architecture général du backend de notre application est structuré sous forme de package selon leur role:

- **Le package security** contient les classes de configuration pour le projet, telles que la configuration de sécurité, la authentification et la configuration Web.

- Le **package controllers** contient les contrôleurs de l'API, qui sont responsables de l'écoute des requêtes HTTP et de la gestion des réponses HTTP.
- Le **package repository** contient les objets de transfert de données, qui sont importés dans controllers et sont utilisés pour transférer des données entre les couches de l'application.
- Le **package exception** contient les exceptions personnalisées, qui sont levées lorsque des erreurs se produisent dans l'application.
- Le **package models** contient les entités de la base de données, qui sont utilisées pour représenter les données dans la base de données.
- Le **package repository** contient les interfaces de repository, qui sont utilisées pour accéder aux données dans la base de données.
- Le **package service** contient les services métier, qui contiennent la logique métier de l'application et communiquent avec les repositories pour accéder aux données.
- Le **package payload** définit les classes pour les objets Request et Response.

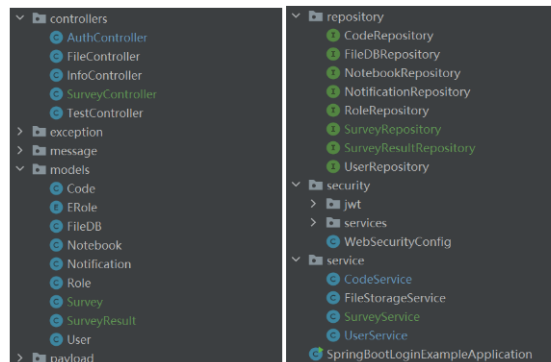


Figure 17: Architecture du backend

Spring Boot fournit déjà des outils pour la gestion des dépendances, la configuration, la gestion des transactions, la sécurité et l'accès aux bases de données

Dans notre projet, on utilise spécifiquement:

- Spring JPA (Java Persistence API) est une bibliothèque qui facilite l'accès aux bases de données relationnelles en utilisant l'ORM (Object-Relational Mapping) en Java. Spring JPA fournit une couche d'abstraction pour communiquer avec la base de données, permettant ainsi aux développeurs de se concentrer sur la logique métier plutôt que sur la gestion de la base de données.
- Spring RestController est une classe qui permet de créer des API RESTful pour exposer des données et des fonctionnalités aux clients via des points de terminaison HTTP pour effectuer des opérations CRUD (Create, Read, Update, Delete) sur des ressources. Il facilite la séparation des responsabilités et la gestion de la logique métier en offrant une abstraction entre les points de terminaison HTTP et les fonctions de traitement de données.
- On utilise également SpringSecurity avec JWT(JSON Web Token) est un moyen pratique pour sécuriser une API REST exposée. JWT est populaire pour l'authentification et l'échange d'informations. Le serveur encode les données dans un jeton Web JSON et l'envoie au client. Le client enregistre le JWT, puis chaque demande du client vers des routes ou des ressources protégées doit être jointe à ce JWT (généralement à l'en-tête). Le serveur validera ce JWT et renverra la réponse.

Voici la structure simple de Spring Security.

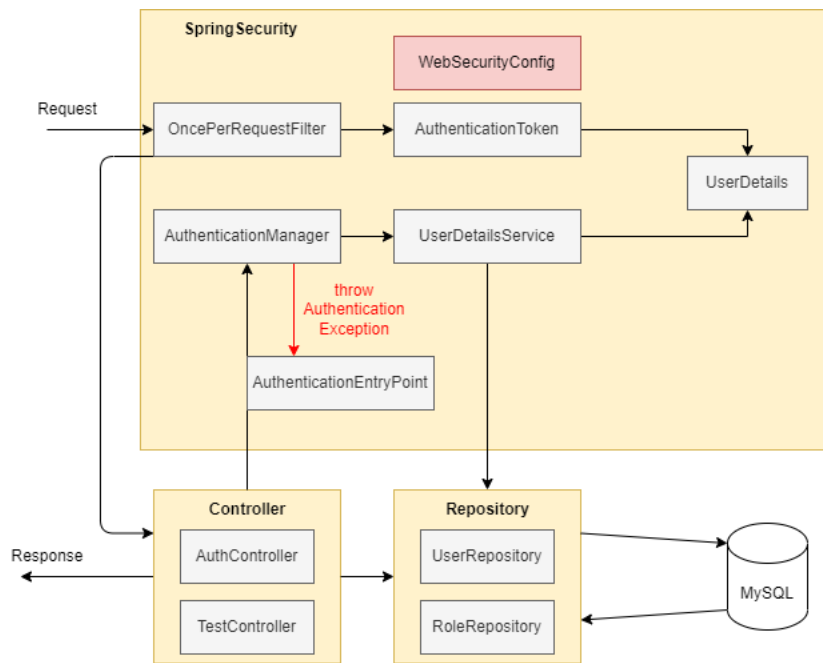


Figure 18: Architecture de serveur d'API Spring Boot Rest avec Spring Security

- **OncePerRequestFilter** effectue une seule exécution pour chaque requête adressée à notre API. Il fournit une méthode `doFilterInternal()` que nous allons implémenter pour analyser et valider JWT, charger les détails de l'utilisateur (à l'aide de `UserDetailsService`), vérifier l'autorisation (à l'aide de `UsernamePasswordAuthenticationToken`).
- **UserDetailsService** a une méthode pour charger l'utilisateur par nom d'utilisateur et renvoie un objet `UserDetails` que Spring Security peut utiliser pour l'authentification et la validation.
- **UserDetails** contient les informations nécessaires (telles que : nom d'utilisateur, mot de passe, autorités) pour créer un objet d'authentification.
- **AuthenticationToken** obtient username, password à partir de la demande de connexion, **AuthenticationManager** l'utilisera pour authentifier un compte de connexion et en cas de succès renvoyer un objet `Authentication` entièrement rempli.
- **AuthenticationEntryPoint** détectera l'erreur d'authentification.
- **Repository** contient `UserRepository` et `RoleRepository` pour fonctionner avec la base de données, sera importé dans le contrôleur.
- **Controller** reçoit et traite la demande après qu'elle a été filtrée par `OncePerRequestFilter`.

Le projet est basé sur l'objectif des utilisateurs de se connecter à leurs propres comptes et d'accéder à leurs propres ressources associées, donc un organigramme uml 19 peut aider à clarifier ce concept.

### 4.3 La couche de persistance des données

Cette couche est responsable de la gestion de l'interaction avec la base de données, où sont stockées les données de l'application. On a utilisé MySQL qui est une base de données relationnelle open source très populaire. Elle est largement utilisée pour stocker des données structurées telles que des informations d'utilisateur, des transactions et des rapports, ainsi que pour les applications web qui nécessitent des interactions en temps réel avec une base de données.

On va décrire via la figure 20 qui suit le modèle de base de donnée.



Table 6: API

Méthode	URL	Actions
POST	/api/auth/signup	créer un nouveau compte pour l'utilisateur
POST	/api/auth/signup_admin	créer un nouveau compte pour l'évaluateur
POST	/api/auth/signin	se connecter à un compte
POST	/api/auth/signout	déconnecter le compte
GET	/api/test/user	accéder au contenu de l'utilisateur
POST	/api/auth/code/upload	déposer un fichier par code
GET	/api/auth/user_id/files	obtenir la liste des fichiers (nom et URL) d'un utilisateur
GET	/api/auth/download/files	télécharger un zip de tous les fichiers
DELETE	/api/auth/files	supprimer tous les fichiers
DELETE	/api/auth/users/id	supprimer un utilisateur
DELETE	/api/auth/admin/id	supprimer un administrateur
GET	/api/info/user_id/codes	obtenir la liste des codes d'un utilisateur
GET	/api/info/users	obtenir la liste des utilisateurs
POST	/api/auth/survey/save	enregistrer un questionnaire
GET	/api/auth/survey/survey_id	obtenir un questionnaire
GET	/api/auth/survey/code/surveycode	obtenir un questionnaire par un code
POST	/api/auth/survey/results	enregistrer les résultats du questionnaire
GET	/api/auth/survey/survey_result	télécharger un zip de tous les résultats du questionnaire
DELETE	/api/auth/survey/survey_result	supprimer les questionnaires

## 5 Gestion de projet

Comme marqué dans la présentation du projet, ce projet dure 8 mois et est divisé en plusieurs itérations de 4 semaines. Des réunions régulières ont été tenues et se tiendront avec les encadrants afin de permettre un bon suivi du projet que ce soit avec l'encadrant entreprise ou l'encadrant Polytech.

Un compte rendu hebdomadaire sera donné à chaque fin de semaine pour voir l'avancement du projet. Un canal de discussion MatterMost a été mis en place pour faciliter les échanges avec les responsables et l'équipe. On pourra y poser des questions, partager les différents documents, rendre les rapports hebdomadaires.

Pour la planification, on a réalisé un Gant pour chaque itération qu'on a mis à jours selon les évènement et l'avancement:

NB. Convention couleur: Rose(Mamisoa) Bleu(Chenrui) Vert(Xianxiang) Rouge(Commun)

		Itération 1	28/11 - 25/12			
	Charge	Semaine 1	Semaine 2	Semaine 3	Semaine 4	Statut
Remarque				Vacance scolaire	Vacance scolaire	
<b>Rectification du cahier de charge</b>	3h					
Enrichissement de l'exploration technologique	2h					
<b>Conception</b>	5h					
Prototypage version 1	5h					
Backend - Squelette du projet Spring Boot	3h					
FrontEnd - Squelette du site via Angular	3h					
Prise en main des frameworks	8h					
Difficultés	<ul style="list-style-type: none"> <li>Beaucoup de rectification au niveau du cahier de charge après la soutenance</li> <li>On a du repenser aux manières de répondre aux besoins des utilisateurs et revoir notre compréhension du besoin de l'utilisateur</li> <li>On a sous estimé le travail derrière la conception</li> <li>Prise en main des frameworks assez laborieux au début</li> </ul>					

Figure 21: Gant mis à jour de l'itération 1

		Itération 2	03/01- 21/01			
	Charge	Semaine 1	Semaine 2	Semaine 3	Semaine 4	Statut
Remarque		Partiel				
Rectification prototype après chaque réunion	3h					
Refaire le prototype en fonction charte graphique	5h					
Rectification conception des fonctionnalités - Modélisation UML	3h					
Modèle conceptuelle de donnée	4h					
Création de la base de donnée MySQL	2h					
Liaison Backend et Base de donnée	3h					
Développement frontEnd de l'interface Aidant (changement après changement de prototype)	5h					
Consulter les fichiers	2h					
Interface Evalueur -Page d'accueil	5h					
Connexion + Authentification	4h					
Questionnaire + documentation sur survey.js	6h					
Difficultés	<ul style="list-style-type: none"> <li>Le fonctionnement en itération est encore nouveau pour nous</li> <li>On a du refaire le prototype après qu'on ait eu une vision de la charte graphique utilisé pour le carnet de bord physique</li> </ul>					

Figure 22: Gant mis à jour de l'itération 2



	Charge	Itération 3	23/01 - 17/02			
Heure affecté au P-trans / emploi du temps		Semaine 1	Semaine 2	Semaine 3	Semaine 4	Status
Remarque			Hyblab	Hyblab		
Modification prototype	2h					
Rectification de l'user story	2h					
Rédaction Livrable 2	7h					
Rectification de l'interface	5h					
Ajout fichier photo /vidéo/audio par l'aidant	3h					
Consulter les événements ajoutés par les aidants	3h					
Trier les événements	3h					
Prise en main de survey.js pour les questionnaires	6h					
Création de questionnaire avec survey.js	5h					
Modifier un questionnaire	2h					
Remplir questionnaire avec survey.js	3h					
Liaison avec la base de donnée	6h					
Difficulté	<ul style="list-style-type: none"> <li>On a eu plusieurs projets en parallèle donc on n'a pas pu se concentrer et donné beaucoup d'heure pour le p-trans</li> <li>Retard au niveau des fonctionnalités de l'itération 2</li> <li>Problème avec la liaison du Backend avec la base de donnée</li> </ul>					

Figure 23: Gant mis à jour de l'itération 3

		Itération 4	20/02 - 17/03			
		Semaine 1	Semaine 2	Semaine 3	Semaine 4	Statut
Remarque		Vacance scolaire	Soutenance	ICreate	ICreate	
Changement de mot de passe à la première connexion	2h					
Changement de mot de passe	2h					
Interface plateforme direct	3h					
Accès direct à un ajout de Fichier via un code	5h					
Accès direct à une formulaire via un code	5h					
Relier le site avec appareil photo de l'appareil	3h					
Prendre photo/vidéo directement sur le site	1h					
Enregistrer audio directement sur le site	1h					
Création et envoi notification	2h					
Statistique direct relié aux activités des aidants	2h					
Livrable 2	5h					
Préparation de la soutenance	7h					

Figure 24: Gant mis à jour de l'itération 4

	Itération 5	20/03 - 16/04			
	Semaine 1	Semaine 2	Semaine 3	Semaine 4	Statut
Remarque					
Création d'une liste de code à la création d'un aidant					ok
Classe les données par aidant dans l'interface Evalueur					ok
Voir les informations des aidants dans l'interface Evalueur					ok
Inscription d'un aidant par l'évaluateur					ok
Inscription d'un évaluateur par un évaluateur					ok
Gestion des aidants					ok
Téléchargement sous forme zip des données					ok
Remise en forme du frontend					ok

Figure 25: Gant mis à jour de l'itération 5

	Itération 6	17/04 - 13/05			
Heure affecté au P-trans / emploi du temps	Semaine 1	Semaine 2	Semaine 3	Semaine 4	Status
Remarque		Vacance scolaire			
Rédaction Livrable Finale					ok
Préparation soutenance					ok
Test					ok
Connexion au serveur					ok
Intégration et déploiement du site					pas ok
Accès direct avec des codes pour questionnaire					ok
Enregistrement questionnaire dans BDD					ok
Mise à jour frontend					ok
Remplir questionnaire avec survey.js					ok
Nettoyage du code					ok

Figure 26: Gant mis à jour de l'itération 6

## 6 Bilan entre ce qui a été demandé et ce qui a été fait

Compte tenu de ce qui a été demandé au début du projet dans le cahier des charges ainsi que les différents ajustements lors des différentes itérations, les principaux besoins du client étaient:

- un plateforme numérique où les aidants peuvent envoyer des ressources numériques de type photo, vidéo ou audio
- un système pour créer un questionnaire pour ensuite être rempli par les aidants.
- réputation des ajouts et des réponses aux questionnaires par les évaluateurs.
- un système d'accès directe par code afin de faciliter l'envoi des fichiers et la réponse aux questionnaires pour les aidants

Nous pouvons constater avec le statut des cas des tests que tous les principaux besoins du client ont été répondu en général. Cependant, il y a quelques fonctionnalités qui auraient été intéressants de développer mais qui n'ont pas abouti. Nous détaillerons plus dans la partie évolution.

Ci-dessous, nous allons montrer via le cas d'utilisation du cahier des charges ,fig 27 et fig 28 ce qui ont été réalisé, marqué en vert et ce qui ne sont pas réalisé, en rouge.

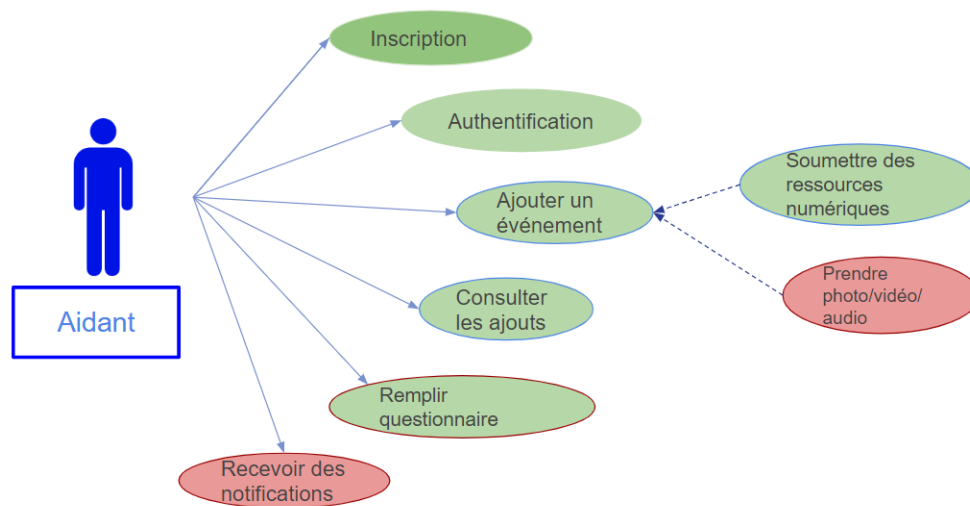


Figure 27: Bilan des réalisations coté Aidant

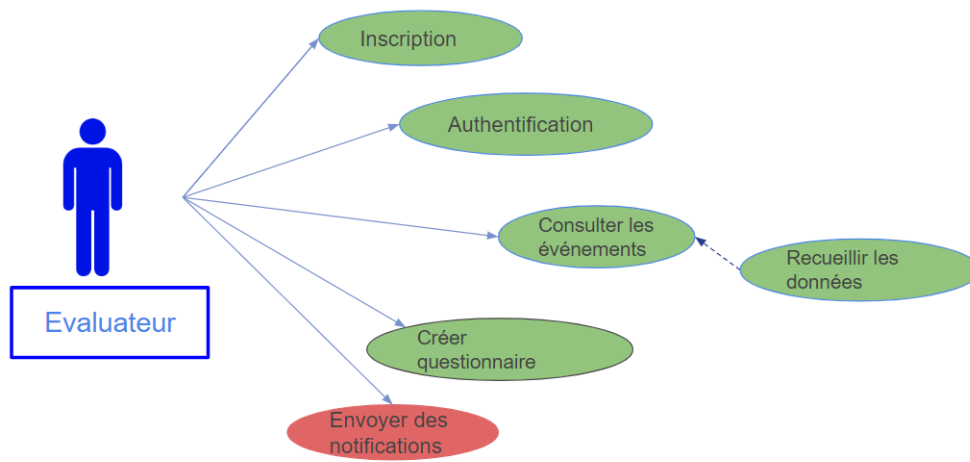


Figure 28: Bilan des réalisations coté Evaluateur

## 7 Conclusion

Durant ce projet transversal en lien avec DIVADOM, on a pu participer à la création d'une plateforme numérique pour le recueil des expériences des aidants. Ce projet nous a permis de monter en compétence en terme de conception, de développement applicatif mais aussi en matière de gestion de projet, relation avec le client, communication et travail d'équipe.

On a pu apprendre à utiliser de frameworks et des systèmes de gestions de bases de données notamment Angular, SpringBoot, MySQL qui sont des technologies mise en avant dans le monde de l'entreprise et qui permettent de nous démarquer et d'enrichir nos compétences technologiques.

On a également pu mettre en tension la conception et la réalisation car la plupart du temps, on a eu des projets où on devait concevoir et non développé ou au contraire, on développe quelque chose qui ont déjà été conçu.

On a pu nous rendre compte de la difficulté de concevoir et de juger la faisabilité en prenant en compte les différents paramètres notamment le temps, la difficulté, les différents parties prenantes et la charge de travail derrière.

Le plus gros gain est que on comprend comment communiquer avec les clients et on se mettons autant que possible à leur place pour répondre aux besoins des clients. Ceci est crucial pour nous pour entrer dans l'entreprise à l'avenir et pour le développement et l'itération d'éventuels futurs produits.

Mais en même temps, DIVANUM a encore de nombreuses limites à ce stade. Nous prenons donc également cela pour donner d'autres évolutions potentielles:

- Améliorer le design, l'interface et les fonctionnalités afin de prendre plus en compte le type d'utilisateur car on n'a pas pu approfondir sur le point de vue de l'utilisateur final. On aurait du prendre contact avec des vrais utilisateurs et effectué des tests UI/UX afin d'adapter notre solution.
- Compléter et enrichir l'interface de l'aidant car même si elle a été créée, il n'y a pour l'instant aucun intérêt d'utilisation étant donnée que tout peut être fait via l'accès directe. Il serait intéressant de mettre des fonctionnalités afin que l'aidant puisse faire des modifications sur ses ajouts, recevoir des notifications, et avoir plus de contrôle sur son compte.
- Mettre en place la liaison entre le DIVA'NUM et l'appareil photo/micro pour la prise de photo/vidéo/audio via l'application Web car n'ayant pas eu de serveur sécurisé plus tôt, on n'a pas pu tester la fonctionnalité qui a besoin d'un serveur sécurisé en https.
- Rendre le serveur parfaitement compatible avec ce produit et mettre-le dans la vie réelle pour mieux servir a les clientes.
- Essayer de rendre le code plus concis pour faciliter le développement ultérieur ou l'utilisation secondaire. Dans le même temps, l'amélioration de la structure de la base de données peut également être un point clé pour faciliter l'intégration du code.
- Pour améliorer la sécurité du code, l'acquisition de données peut être davantage cryptée.

## 8 Annexe

### 8.1 Contexte technologique

Dans cette partie, on a commencé à faire une exploration technologique afin de choisir les différentes technologies qu'on va utiliser étant donné que ce n'est pas imposé par le client. Voici alors le résultat de nos recherches qui est encore susceptible d'être modifié avant le commencement du développement.

- Mattermost pour la communication entre les différents membres du projet.
- Google Doc pour la rédaction collaborative des documents textuels.
- Latex pour les rendus finaux.
- Draw.io pour les différents diagrammes.
- Figma pour le prototype/maquette.
- Xtensio pour la persona.
- Gitlab pour la gestion de version du code.
- Angular: framework utilisé pour la partie FrontEnd.
- Java SpringBoot: framework pour la partie BackEnd.
- MySql pour la base de donnée

#### Outil de communication

**Mattermost:** un logiciel de messagerie instantané conseillé et utilisé par nos tuteurs enseignants.

#### Modélisation et design

**draw.io:** il permet de dessiner les différents diagrammes pour la modélisation notamment le diagramme de cas d'utilisation, le modèle de domaine, le diagramme de classe ainsi que le diagramme de séquence

**Xtensio:** est un logiciel permettant de créer des personas qui sont des personnages fictifs représentant les utilisateur-types de Diva'Num

**Figma:** c'est un éditeur de graphiques vectoriels et un outil de prototypage. Il sera utilisé pour concevoir le design de Diva'Num. Il permet aussi un travail d'équipe à distance, instantané.

#### Rédaction rapport et rendu

**Google doc:** Du fait de sa facilité d'utilisation et de sa popularité, on a choisi google doc comme outil de rédaction collaboratif entre tous les membres du projet. On pourra ainsi chacun ajouter, éditer et commenter le travail.

**Latex:** système de composition du rendu final, il permet de se concentrer sur le contenu du document sans se soucier de la mise en forme qui sera effectuée automatiquement.

#### Gestion de version

##### Gitlab

On utilise git pour la gestion des versions du code. En effet, elle permet de faciliter la collaboration en permettant de revenir sur les anciennes versions. La plateforme GitLab va contenir quant à elle les dépôts dans le cloud afin d'arriver à travailler sur un même projet et distinguer en temps réel les modifications apportées par les autres développeurs.

#### Serveur

Pour la question du serveur qui va héberger l'application Web, pour l'instant, on utilise nos propres ordinateurs

portables comme serveur pour tester, ensuite on va demandé l'accord pour l'utilisation de ceux de l'école jusqu'à ce que DIVADOM trouve une solution plus concrète.

## References

- [1] [Building a Web Application with Spring Boot and Angular](#)
- [2] [Débuter avec Angular](#)
- [3] [SurveyJS](#)
- [4] [JPA Query](#)
- [5] [MYSQL](#)